

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2006-107126

(P2006-107126A)

(43) 公開日 平成18年4月20日(2006.4.20)

(51) Int. Cl.	F I	テーマコード (参考)
<b>G06F 3/06 (2006.01)</b>	G06F 3/06 304P	5B014
<b>G06F 12/00 (2006.01)</b>	G06F 12/00 501B	5B065
<b>G06F 13/10 (2006.01)</b>	G06F 12/00 514E	5B082
	G06F 12/00 545A	
	G06F 13/10 340A	
審査請求 未請求 請求項の数 15 O L (全 40 頁)		

(21) 出願番号 特願2004-292849 (P2004-292849)  
 (22) 出願日 平成16年10月5日 (2004.10.5)

(71) 出願人 000005108  
 株式会社日立製作所  
 東京都千代田区丸の内一丁目6番6号  
 (74) 代理人 100075513  
 弁理士 後藤 政喜  
 (74) 代理人 100084537  
 弁理士 松田 嘉夫  
 (74) 代理人 100114236  
 弁理士 藤井 正弘  
 (72) 発明者 越智 啓之  
 神奈川県横浜市戸塚区戸塚町5030番地  
 株式会社日立製作所ソフトウェア事業部  
 内

最終頁に続く

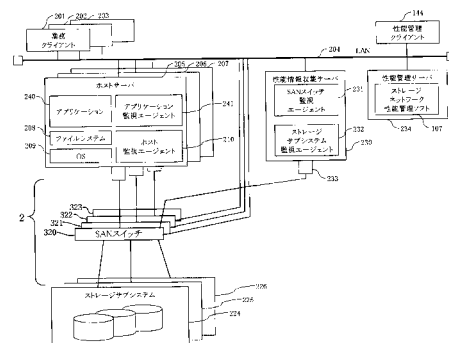
(54) 【発明の名称】 ストレージネットワーク性能情報の収集・保存方法及び計算機システム並びにプログラム

(57) 【要約】

【課題】アプリケーションの性能劣化の原因を確実に追究可能にするストレージネットワーク性能情報の収集・保存方法を提供する。

【解決手段】ホストサーバ205及びストレージ装置224を含むストレージネットワーク2の構成要素から収集・保存した性能情報を、アプリケーション240の性能劣化の調査に用いられる可能性の有無を自動的に判別し、この判別結果に応じて重要度を決定する。そして、重要度が高い性能情報の保存期間を長く設定し、重要度の低い性能情報の保存期間を短く設定する。そして、性能情報毎に設定された保存期間が過ぎると、該当する性能情報を削除する。

【選択図】 図1



**【特許請求の範囲】****【請求項 1】**

計算機とストレージ装置との間で入出力データを送受信するネットワーク装置を有するストレージネットワークのリソースについて性能情報を収集し、保存するストレージネットワーク性能情報の収集・保存方法であって、

前記ストレージネットワークとストレージ装置及び計算機のリソースのうち、予め設定した収集対象要素から性能情報を収集し、保存する性能情報収集ステップと、

前記収集対象要素間に成り立つ性能の依存関係を収集し、保存する性能依存関係収集ステップと、

前記計算機で実行されるアプリケーションの性能劣化期間を検出し、当該性能劣化期間を保存するアプリケーション性能劣化期間検出ステップと、

前記検出したアプリケーション性能劣化期間中に、性能劣化を起こしたアプリケーションと性能の依存関係にあった収集対象要素を検出する性能劣化時依存関係検出ステップと

、

前記性能劣化時依存関係検出ステップで検出された収集対象要素の性能情報について通常の保存期間よりも延長した第2の保存期間を設定し、前記抽出対象外の性能情報について通常の保存期間を設定するステップと、

前記通常または第2の保存期間が過ぎた性能情報を削除する削除ステップと、を含むことを特徴とするストレージネットワーク性能情報の収集・保存方法。

**【請求項 2】**

前記性能劣化時依存関係検出ステップは、前記収集対象要素間の性能の依存関係に基づいて、前記性能劣化を起こしたアプリケーションを起点として、当該性能劣化が起きた期間中に、当該アプリケーションが性能上の負荷をかけるストレージネットワークの下流側に前記依存関係を辿った経路上の収集対象要素を、当該アプリケーションと性能の依存関係にあった収集対象要素とすることを特徴とする請求項1に記載のストレージネットワーク性能情報の収集・保存方法。

**【請求項 3】**

前記性能劣化時依存関係検出ステップは、前記収集対象要素間の性能の依存関係に基づいて、前記性能劣化を起こしたアプリケーションを起点として、当該性能劣化が起きた期間中に、当該アプリケーションが性能上の負荷をかけるストレージネットワークの下流側に前記依存関係を辿った経路上の収集対象要素を検出し、検出された各収集要素から見て性能上の負荷をかける上流側に前記依存関係を辿った経路上の収集要素を、当該アプリケーションと性能の依存関係にあった収集対象要素とすることを特徴とする請求項1に記載のストレージネットワーク性能情報の収集・保存方法。

**【請求項 4】**

前記性能劣化時依存関係検出ステップは、前記収集対象要素間の性能の依存関係に基づいて、前記性能劣化を起こしたアプリケーションを起点として、当該性能劣化が起きた期間中に、当該アプリケーションが性能上の負荷をかけるストレージネットワークの下流側に前記依存関係を辿った経路上の収集対象要素を検出し、検出された各収集要素から見て性能上の負荷をかける下流側に前記依存関係を辿った経路上の収集対象要素を、当該アプリケーションと性能の依存関係にあった収集対象要素とすることを特徴とする請求項1に記載のストレージネットワーク性能情報の収集・保存方法。

**【請求項 5】**

前記性能劣化時依存関係検出ステップは、前記収集対象要素間の性能の依存関係に基づいて、前記性能劣化を起こしたアプリケーションを起点として、当該性能劣化が起きた期間中に、当該アプリケーションが性能上の負荷をかけるストレージネットワークの下流側に前記依存関係を辿った経路上の収集対象要素を検出し、検出された各収集要素から見て性能上の負荷をかける下流側及び上流側に前記依存関係を辿った経路上の収集対象要素を、当該アプリケーションと性能の依存関係にあった収集対象要素とすることを特徴とする請求項1に記載のストレージネットワーク性能情報の収集・保存方法。

10

20

30

40

50

## 【請求項 6】

計算機とストレージ装置との間で入出力データを通信するネットワーク装置を有するストレージネットワークのリソースについて性能情報を収集し、保存するプログラムであって、

前記ストレージネットワークとストレージ装置及び計算機のリソースのうち、予め設定した収集対象要素から性能情報を収集し、保存する処理と、

前記収集対象要素間に成り立つ性能の依存関係を収集し、保存する処理と、

前記計算機で実行されるアプリケーションの性能劣化期間を検出し、当該性能劣化期間を保存する処理と、

前記検出したアプリケーション性能劣化期間中に、性能劣化を起こしたアプリケーションと性能の依存関係にあった収集対象要素を検出する性能劣化時依存関係検出処理と、 10

前記性能劣化時依存関係検出処理で検出された収集対象要素の性能情報について通常の保存期間よりも延長した第 2 の保存期間を設定し、前記抽出対象外の性能情報について通常の保存期間を設定する処理と、

前記通常または第 2 の保存期間が過ぎた性能情報を削除する削除処理と、  
をコンピュータに機能させることを特徴とするプログラム。

## 【請求項 7】

前記性能劣化時依存関係検出処理は、前記収集対象要素間の性能の依存関係に基づいて、前記性能劣化を起こしたアプリケーションを起点として、当該性能劣化が起きた期間中に、当該アプリケーションが性能上の負荷をかけるストレージネットワークの下流側に前記依存関係を辿った経路上の収集対象要素を、当該アプリケーションと性能の依存関係にあった収集対象要素とすることを特徴とする請求項 6 に記載のプログラム。 20

## 【請求項 8】

前記性能劣化時依存関係検出処理は、前記収集対象要素間の性能の依存関係に基づいて、前記性能劣化を起こしたアプリケーションを起点として、当該性能劣化が起きた期間中に、当該アプリケーションが性能上の負荷をかけるストレージネットワークの下流側に前記依存関係を辿った経路上の収集対象要素を検出し、検出された各収集要素から見て性能上の負荷をかける上流側に前記依存関係を辿った経路上の収集要素を、当該アプリケーションと性能の依存関係にあった収集対象要素とすることを特徴とする請求項 6 に記載のプログラム。 30

## 【請求項 9】

前記性能劣化時依存関係検出処理は、前記収集対象要素間の性能の依存関係に基づいて、前記性能劣化を起こしたアプリケーションを起点として、当該性能劣化が起きた期間中に、当該アプリケーションが性能上の負荷をかけるストレージネットワークの下流側に前記依存関係を辿った経路上の収集対象要素を検出し、検出された各収集要素から見て性能上の負荷をかける下流側に前記依存関係を辿った経路上の収集対象要素を、当該アプリケーションと性能の依存関係にあった収集対象要素とすることを特徴とする請求項 6 に記載のプログラム。

## 【請求項 10】

前記性能劣化時依存関係検出処理は、前記収集対象要素間の性能の依存関係に基づいて、前記性能劣化を起こしたアプリケーションを起点として、当該性能劣化が起きた期間中に、当該アプリケーションが性能上の負荷をかけるストレージネットワークの下流側に前記依存関係を辿った経路上の収集対象要素を検出し、検出された各収集要素から見て性能上の負荷をかける下流側及び上流側に前記依存関係を辿った経路上の収集対象要素を、当該アプリケーションと性能の依存関係にあった収集対象要素とすることを特徴とする請求項 6 に記載のプログラム。 40

## 【請求項 11】

アプリケーションを実行する計算機と、

前記計算機とストレージ装置との間で入出力データを送受信するネットワーク装置を含むストレージネットワークと、を備えた計算機システムにおいて、 50

前記計算機とストレージ装置及びストレージネットワークのリソースのうち、予め設定した収集対象から性能情報を収集し、この性能情報を予め設定した記憶領域へ格納する性能情報収集部と、

前記収集対象要素間に成り立つ性能の依存関係を前記ストレージネットワークの構成に基づいて収集し、当該収集した性能の依存関係を保存する性能依存関係収集部と、

前記アプリケーションの性能の劣化を検出し、当該アプリケーションの性能劣化期間を保存する劣化検出部と、

前記アプリケーションの性能劣化期間中に、性能劣化を起こしたアプリケーションと性能の依存関係にあった収集対象要素を検出する性能劣化時依存関係検出部と、

前記性能劣化時依存関係検出部で検出された収集対象要素の性能情報について通常の保存期間よりも延長した第2の保存期間を設定し、前記抽出対象外の性能情報について通常の保存期間を設定する保存期間設定部と、

前記通常または第2の保存期間が過ぎた性能情報を検出し、前記記憶領域から削除する削除部と、

を備えたことを特徴とする計算機システム。

#### 【請求項12】

前記性能劣化時依存関係検出部は、前記収集対象要素間の性能の依存関係に基づいて、前記性能劣化を起こしたアプリケーションを起点として、当該性能劣化が起きた期間中に、当該アプリケーションが性能上の負荷をかけるストレージネットワークの下流側に前記依存関係を辿った経路上の収集対象要素を、当該アプリケーションと性能の依存関係にあ

#### 【請求項13】

前記性能劣化時依存関係検出部は、前記収集対象要素間の性能の依存関係に基づいて、前記性能劣化を起こしたアプリケーションを起点として、当該性能劣化が起きた期間中に、当該アプリケーションが性能上の負荷をかけるストレージネットワークの下流側に前記依存関係を辿った経路上の収集対象要素を検出し、検出された各収集要素から見て性能上の負荷をかける上流側に前記依存関係を辿った経路上の収集対象要素を、当該アプリケーションと性能の依存関係にあった収集対象要素とすることを特徴とする請求項11に記載の計算機システム。

#### 【請求項14】

前記性能劣化時依存関係検出部は、前記収集対象要素間の性能の依存関係に基づいて、前記性能劣化を起こしたアプリケーションを起点として、当該性能劣化が起きた期間中に、当該アプリケーションが性能上の負荷をかけるストレージネットワークの下流側に前記依存関係を辿った経路上の収集対象要素を検出し、検出された各収集要素から見て性能上の負荷をかける下流側に前記依存関係を辿った経路上の収集対象要素を、当該アプリケーションと性能の依存関係にあった収集対象要素とすることを特徴とする請求項11に記載の計算機システム。

#### 【請求項15】

前記性能劣化時依存関係検出部は、前記収集対象要素間の性能の依存関係に基づいて、前記性能劣化を起こしたアプリケーションを起点として、当該性能劣化が起きた期間中に、当該アプリケーションが性能上の負荷をかけるストレージネットワークの下流側に前記依存関係を辿った経路上の収集対象要素を検出し、検出された各収集要素から見て性能上の負荷をかける下流側及び上流側に前記依存関係を辿った経路上の収集対象要素を、当該アプリケーションと性能の依存関係にあった収集対象要素とすることを特徴とする請求項11に記載の計算機システム。

#### 【発明の詳細な説明】

#### 【技術分野】

#### 【0001】

本発明は、ストレージネットワークを構成するハードウェア装置ならびに同装置で稼動するソフトウェアに関する性能情報を収集し、保存する方法に係り、特にネットワークが

10

20

30

40

50

大規模となり性能情報の収集対象となる構成要素の数が膨大になる場合に好適なストレージネットワーク性能情報の収集・保存方法に関する。

【背景技術】

【0002】

ネットワークを經由して複数のホストサーバから統合化されたストレージ装置にアクセスする構成をとるストレージネットワーク(SAN)は、大規模化が進むストレージの利用効率を高め管理コストを削減するためのデータセンタ用アーキテクチャとして広く普及しつつある。このようなストレージネットワーク環境(以下、SAN環境)で業務システムの性能監視やチューニングを行うには、ネットワークを構成するさまざまなハードウェア装置ならびにソフトウェアに関する性能情報を包括的に収集して、その相互間の関係や時間的な推移を把握することが必要になる。

10

【0003】

計算機と外部記憶装置が直結されたサーバ単位で各業務システムが独立していた従来のアーキテクチャと異なり、SAN環境ではネットワーク装置やストレージ装置等を複数の計算機で共有する。SANの共用部分では各計算機で実行される業務システム間で性能上の干渉が生じる可能性がある。このため性能情報を包括的に収集して、計算機、ネットワーク装置、ストレージ装置の相互間の関係や時間的な性能の推移を把握する。

【0004】

SANを対象にした従来の性能管理ソフトウェアでは、性能監視対象となるハードウェア装置やソフトウェアごとにネットワーク内に配置したエージェントと、ネットワーク全体の性能情報を一元管理する管理ソフト(ストレージ管理ソフトウェア)からなる構成をとって上記要求に込えている。

20

【0005】

各エージェントはそれぞれの監視対象と直接通信して性能情報を取得し、一方、管理ソフトウェアは、各エージェントが取得した性能情報を収集・保存し、ストレージネットワーク管理者等からの要求に込じて性能情報を提供する。

【0006】

SANを基盤にしたストレージ統合の普及とともに、大規模化したネットワークの構成要素(リソース)の数が膨大となり、その相互関係も複雑化する傾向にある。このリソースの増大と複雑化に伴い、ストレージ管理ソフトが保存対象とする性能情報も膨大になる。しかし、ストレージ装置の限られた記憶容量内で、効果的に性能情報を保存するため、次のような方法が採られている。

30

【0007】

方法1:一度収集した性能情報を、ストレージ管理者が指定した一定期間のみ保存し、一定期間が過ぎると削除する方法。

【0008】

方法2:保存容量の上限値を決め、上限値を超える都度、過去の性能情報から削除する方法。

【0009】

方法3:収集した粒度の細かい性能情報を、より粒度の荒い値に時系列方向に集約し、粒度の荒い性能情報ほど長期間保存する方法。例えば、毎分の性能情報を60回取得した後、それらの値の平均値といった統計情報を計算し、毎時の性能情報を得る。毎分の性能情報に比べ粒度の荒い毎時の性能情報を長期間保存する。なお、粒度とは、単位時間当たりの性能情報のサンプル数を示し、単位時間当たりのサンプル数が多ければ粒度は細くなり、単位時間当たりのサンプル数が少なければ粒度は荒くなる。

40

【0010】

上記方法3を、ネットワークのトラフィックデータに対して適用した例として、特許文献1が知られている。

【特許文献1】米国特許第5231593号

【発明の開示】

50

**【発明が解決しようとする課題】****【0011】**

SANに対するI/O実行を行うアプリケーションに性能劣化が発生した場合、SANの管理者は、性能劣化の原因を調査する必要がある。性能劣化原因の調査は、ストレージ管理ソフトウェアが保存している性能情報のうち、アプリケーションが性能劣化を起こした時刻前後に、アプリケーションの発行するI/O命令の実行に関わった装置の性能情報を閲覧することで行う。性能劣化の原因調査には、粒度の細かい性能情報が必要とされる。

**【0012】**

しかしながら、アプリケーションに性能劣化が起きてから、性能劣化が発覚し、SAN管理者が性能劣化原因の調査を開始するまでに、数日のブランクがあることがある。このような場合には、上記した方法1~3を用いても、調査開始時には粒度の細かい性能情報が消去されていることがあり、性能劣化原因を追究できない、という問題があった。

**【0013】**

そこで、本発明は、アプリケーションの性能劣化の原因を確実に追究可能にするストレージネットワーク性能情報の収集・保存方法を提供することを目的とする。

**【課題を解決するための手段】****【0014】**

本発明は、計算機及びストレージ装置を含むストレージネットワークの構成要素から収集・保存した性能情報を、アプリケーションの性能劣化の調査に用いられる可能性の有無を自動的に判別し、この判別結果に応じて重要度を決定する。そして、重要度が高い性能情報の保存期間を長く設定し、重要度の低い性能情報の保存期間を短く設定する。そして、性能情報毎に設定された保存期間が過ぎると、該当する性能情報を削除する。

**【発明の効果】****【0015】**

したがって、本発明は、ストレージネットワークに対しI/O実行を行うアプリケーションで性能劣化が発生した場合、性能劣化の原因調査に必要な性能情報を自動的に絞込み、この絞込結果に限定して所定の時間保存することが可能となり、性能情報を格納するためのデータ記憶容量の増大を抑制しながら、性能劣化の原因追及に必要な性能情報についてのみ長期に渡って保存でき、性能劣化の調査が可能となる。

**【発明を実施するための最良の形態】****【0016】**

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。

**【0017】**

図1は、本発明を適用するシステム構成図である。ローカルエリアネットワーク(LAN)204には、ホストサーバ205~207と業務クライアント201~203が接続される。各業務クライアント201~203は、ホストサーバ205~207上で実行されるアプリケーション240のサービス(業務の提供)を受ける。

**【0018】**

各ホストサーバ205~207は、ストレージネットワーク(SAN)2を介してストレージサブシステム224~226に接続され、データなどの読み書きを行う。SAN2は、例えば、ファイバチャネル等で構成されており、複数のSANスイッチ320~323がホストサーバ205~207とストレージサブシステム224~226に接続される。なお、LAN204には、図示しないSAN管理用のサーバが接続され、各SANスイッチ320~323の設定を行う。このため、LAN204はSANスイッチ320~323にも接続される。

**【0019】**

LAN204には、SAN2とストレージサブシステム224~226の性能情報を収集する性能情報収集サーバ230と、性能情報収集サーバ230及び後述するようにホストサーバの各監視エージェントが収集した性能情報を管理する性能管理サーバ234と、

10

20

30

40

50

性能管理サーバ234を操作するための性能管理クライアント144も接続される。

【0020】

性能情報収集サーバ230は、SAN2とストレージサブシステム224～226の性能を監視するため、SANスイッチ320～323にも接続される。また、性能管理サーバ234は、LAN204だけではなく、SAN2のSANスイッチ320～323のいずれかと接続するポートを有し、ストレージサブシステム224～226へアクセス可能としてもよい。

【0021】

次に、SAN2を介して接続されたホストサーバ205～207とストレージサブシステム224～226を監視して、性能情報を監視、収集する性能管理システムの概要について説明する。 10

【0022】

図1において、ホストサーバ205～207では、それぞれOS209、ファイルシステム208及びアプリケーション240が稼動し、さらにホストサーバ205～207を監視して性能情報を収集するホスト監視エージェント210と、各アプリケーション240を監視して性能情報を収集するアプリケーション監視エージェント241が稼動する。

【0023】

ホスト監視エージェント210は、各ホストサーバ205～207上でそれぞれ実行されて、各ホストサーバ205～207を監視して性能情報を収集する。ホスト監視エージェント210は、収集した性能情報をストレージサブシステム224～226の予め設定された領域に格納する。そして、ホスト監視エージェント210は、性能管理サーバ234との間で通信を行って、後述するように取得した性能情報を、性能管理サーバ234のSAN性能管理ソフトウェア107に提供する。 20

【0024】

ここで、ホスト監視エージェント210が監視する対象としては、例えば、ホストサーバ205～207のファイルシステム208のI/O回数、各ホストサーバ205～207に割り当てられたボリュームのI/O回数、ホストサーバ205～207に設けたSAN2と通信するポートのI/O回数などを性能の監視対象とし、これら監視対象の性能情報を収集し、ストレージシステムの所定の領域に格納する。

【0025】

アプリケーション監視エージェント241は、各ホストサーバ205～207上のアプリケーション240をそれぞれ監視して性能情報を収集するもので、各ホストサーバ205～207上のアプリケーション240毎に実行される。 30

【0026】

アプリケーション監視エージェント241は、収集した性能情報をストレージサブシステム224～226の予め設定された領域に格納する。そして、ホスト監視エージェント210は、性能管理サーバ234との間で通信を行って、SAN性能管理ソフトウェア234に取得した性能情報を提供する。

【0027】

ここで、アプリケーション監視エージェント241が監視する対象としては、例えば、各アプリケーション240のレスポンスタイム(ターンアラウンドタイム)であり、各業務クライアント201～203からの要求を受信してからこの要求に回答するまでのレスポンスタイムを監視する。 40

【0028】

さらに、アプリケーション監視エージェント241は、後述するように、レスポンスタイムが予め設定した値よりも悪化した場合には、アプリケーション240の性能劣化を判定し、性能が劣化した期間を検出する。そして、アプリケーション監視エージェント241は、アプリケーション240の性能が劣化した期間を、ストレージサブシステム224～226の予め設定された領域に格納する。そして、アプリケーション監視エージェント241は、性能管理サーバ234との間で通信を行って、格納した性能情報を提供する。 50

## 【0029】

性能情報収集サーバ230では、SAN2のSANスイッチ320～323を監視して性能情報を収集するSANスイッチ監視エージェント231と、ストレージサブシステム224～226を監視して性能情報を収集するストレージサブシステム監視エージェント232が稼動する。

## 【0030】

SANスイッチ監視エージェント231は、各SANスイッチ320～323をそれぞれ監視して性能情報を収集するものである。

## 【0031】

SANスイッチ監視エージェント231は、収集した性能情報をストレージサブシステム224～226の予め設定された領域に格納する。そして、SANスイッチ監視エージェント231は、性能管理サーバ234との間で通信を行って、取得した性能情報をSAN性能管理ソフトウェア107に提供する。

10

## 【0032】

ここで、SANスイッチ監視エージェント231が監視する対象としては、例えば、各SANスイッチ320～323の単位時間あたりのI/O回数であり、この監視対象の性能情報を収集し、ストレージシステムの所定の領域に格納する。

## 【0033】

ストレージサブシステム監視エージェント232は、各ストレージサブシステム224～226をそれぞれ監視して性能情報を収集するものである。

20

## 【0034】

ストレージサブシステム監視エージェント232は、収集した性能情報をストレージサブシステム224～226の予め設定された領域に格納する。そして、ストレージサブシステム監視エージェント232は、性能管理サーバ234との間で通信を行って、収集した性能情報をSAN性能管理ソフトウェア107に提供する。

## 【0035】

ここで、ストレージサブシステム監視エージェント232が監視する対象としては、例えば、各ストレージサブシステムの単位時間あたりのI/O回数であり、より具体的には、後述するように、各ストレージサブシステムのポートの単位時間あたりのI/O回数、各ボリューム（論理ボリューム）の単位時間あたりのI/O回数、RAIDグループ毎の単位時間あたりのI/O回数等から構成される。

30

## 【0036】

そして、性能管理サーバ234上では、上記各エージェントが収集した性能情報を管理するストレージネットワーク（SAN）性能管理ソフトウェア107が実行される。

## 【0037】

SAN性能管理ソフトウェア107は、後述する図41のように、アプリケーション監視エージェント241が性能の悪化を検出した場合には、該当するアプリケーション240の性能が劣化した期間について、このアプリケーション240に関連するSAN2上のリソースの性能情報について、保存期間を重要データとして延長する。

## 【0038】

なお、性能の劣化に関与しないリソースの性能情報については、通常データとして所定の保存期間を設定する。

40

## 【0039】

そして、通常の保存期間と重要データの保存期間が経過した性能情報について削除を行う。

## 【0040】

つまり、本性能管理システムでは、SAN2上の障害や設定の不具合などを、アプリケーション240の性能情報（レスポンスタイム）から判定し、後述するように性能劣化が生じたアプリケーション240が利用するI/Oパス（経路情報）に係わるSAN2上のリソースの性能情報を、重要データとして通常よりも長い保存期間に設定し、後に障害等

50



の追究が可能とする。

【0041】

一方、アプリケーション240の性能劣化に関連しないSAN2上のリソースの性能情報については、重要データに比して期間の短い通常の保存期間まで保存し、通常の保存期間が経過した後に削除する。

【0042】

これにより、性能情報を格納する記憶領域が膨大になるのを防ぎながらも、アプリケーション240に性能の劣化が生じてから、通常データの保存期間が過ぎた後であっても、アプリケーション240のI/Oパスに関連するリソースについては、粒度の細かい性能情報が保存されているので、アプリケーション240の性能劣化の原因を追究することが可能となるのである。

10

【0043】

図2は、上記図1に示した性能管理システムの機能の概要を示す説明図である。この図2において、ストレージネットワーク構成装置・ソフトは、上記図1に示した、SAN2、ストレージサブシステム224～226、ホストサーバ205～207のハードウェア及びOS、ファイルシステム等のSAN2に関連するリソースを示す。

【0044】

また、装置管理エージェント20は、上記図1のSANスイッチ監視エージェント231、ストレージサブシステム監視エージェント232、ホスト監視エージェント210の総称であり、SAN2上のリソースのそれぞれを監視して性能情報を収集し格納するものである。

20

【0045】

各ストレージネットワーク構成装置・ソフトは、SAN2上の各リソースの性能情報を取得する性能情報取得機能108と、各リソースの構成情報を取得する構成情報取得機能114を持つ。

【0046】

各装置監視エージェント20は、SAN2上のリソースの性能情報を収集する性能情報収集機能109と、収集した性能情報をメトリクス値に変換するメトリクス値テーブル110と、SAN性能管理ソフトウェア107に対して性能情報(メトリクス値)の通知等を行う性能情報応答機能111、SAN2上のリソースから構成情報を取得する構成情報収集機能115、SAN2上のリソースをアプリケーション240のI/Oパスに基づいて、リソース間の関連を記憶するリソース間関連情報記憶部116、取得した構成情報をSAN性能管理ソフトウェア107に回答する構成情報応答機能117を持つ。

30

【0047】

ホストサーバ205～207で実行される各アプリケーション240は、各アプリケーション240が利用するSAN2上のリソースを取得する構成情報取得機能126を持ち、各アプリケーション240の性能(例えば、レスポンスタイム)をログ130に出力する。

【0048】

アプリケーション監視エージェント241は、アプリケーション240が利用するSAN2上のリソースの構成情報を収集する構成情報収集機能127、アプリケーション240が利用するSAN2上のリソースの関連を保存するリソース間関連情報記憶部128、上記構成情報が取得した構成情報をSAN性能管理ソフトウェア107に回答する構成情報応答機能129、ログ130を読み込んでアプリケーション240の劣化を検出するアプリケーション性能劣化期間検出機能131、アプリケーション240の劣化の判定を行うための情報を示すアプリケーション性能劣化期間決定ルール132、上記劣化が検出された期間を性能劣化として記憶するアプリケーション性能劣化期間テーブル134と、アプリケーション性能劣化期間テーブル134の情報をSAN性能管理ソフトウェア107に回答するアプリケーション性能劣化情報応答機能135を持つ。

40

【0049】

50

SAN性能管理ソフト107は、各装置監視エージェント20からSAN2上のリソースの性能情報を収集する性能情報収集機能112と、性能情報として収集したメトリクス値を格納するメトリクス値テーブル119と、SAN2上のリソースの構成情報を収集する構成情報収集機能118と、SAN2上のリソースの関係を記憶するリソース間関連情報記憶部119と、アプリケーション監視エージェント241から性能劣化に関する情報を収集するアプリケーション性能劣化情報収集機能136と、アプリケーション毎に性能劣化期間を保存するアプリケーション性能劣化期間テーブル137と、通常データと重要データの識別を行ってアプリケーションの性能劣化に関する性能情報の保存期間を延長する削除非対象リソース算出機能138及び削除データ制御テーブル139と、保存期間を過ぎたメトリクス値(性能情報)をストレージサブシステム224~226の所定の領域から削除するメトリクス削除機能140と、SAN2上のリソース毎にメトリクス値の保存期間を設定するメトリクス保存期間テーブル141と、アプリケーション240の劣化を判定するための目標値を設定するアプリケーション目標性能設定機能133と、性能情報と収集したメトリクス値の保存期間を設定するメトリクス保存期間設定機能142と、SAN2上のリソースのメトリクス値から性能を分析する性能分析表示機能143を持つ。

10

20

30

40

50

#### 【0050】

アプリケーション監視エージェント241は、各ホストサーバ205~207上で実行され、各ホストサーバのアプリケーション240の性能劣化を検出するソフトウェアである。アプリケーション監視エージェント241は、アプリケーションおよびアプリケーションが利用するファイルシステムとの対応関係も検出する。

#### 【0051】

SAN2の性能情報の収集と監視は以下のように行なわれる。装置監視エージェント21の性能情報収集機能109は、予め設定したスケジューリング設定に従い定期的にタイマーによって起動されるか、SAN性能管理ソフトウェア107の要求で起動されると、自エージェントが担当するストレージネットワーク構成装置・ソフト21が持つ性能情報取得機能108に対し、メトリクスの測定値の送信を要求する。ストレージネットワーク構成要素(リソース)の個々の性能情報で、性能監視の候補となるものをメトリクスと呼ぶ。そして、このメトリクスは収集した性能情報を示し、例えば、種類の異なる各性能情報を相互に比較可能とするため、性能情報を変換したものである。

#### 【0052】

上記装置監視エージェント20からの要求に応じて、SAN2上のリソースの性能情報取得機能108から返信されたメトリック値は、装置監視エージェント20の性能情報収集機能109によってメトリクス値テーブル110に格納される。

#### 【0053】

SAN性能管理ソフトウェア107の性能情報収集機能112も同様に、予め設定したスケジューリング設定に従い定期的に起動され、装置監視エージェント20の性能情報応答機能111に対してメトリクス値の送信を要求する。この要求を受けた装置監視エージェント20は、性能情報応答機能111から要求されたメトリクス値をメトリクス値テーブル110から検索し、SAN性能管理ソフトウェア107の性能情報収集機能112に返信する。SAN性能管理ソフトウェア107は、装置監視エージェント20の性能情報応答機能111から返信されたメトリクス値を、性能情報収集機能112によってメトリクス値テーブル113に格納する。

#### 【0054】

SAN2の構成要素で、ひとまとまりのメトリクス値を取得する単位となるものをリソースと呼ぶ。リソースとリソース間の関連については、後述の図3で説明する。また、SAN性能管理ソフトウェア107の性能分析表示機能143が性能管理クライアント144に表示する画面の具体例を図9で説明する。

#### 【0055】

リソース間の関連情報の収集は、上記性能情報の場合と同様、以下のように行なわれる

。

## 【0056】

装置監視エージェント20の構成情報収集機能115は、予め設定されたスケジューリング設定で定期的に起動されるか、SAN性能管理ソフトウェア107の要求で起動され、自エージェント20が担当するストレージネットワーク構成装置・ソフト21の構成情報取得機能114に対しリソース間関連情報の送信を要求し、要求した情報が返信されるとリソース間の構成情報に基づくリソース間の関連情報をリソース間関連情報記憶部116に格納する。

## 【0057】

SAN性能管理ソフトウェア107の構成情報収集機能118は、予め設定したスケジューリング設定で定期的に起動され、監視エージェント20の構成情報応答機能117に対し各エージェント20が収集したリソース間関連情報の送信を要求し、要求した情報が各装置監視エージェント20のリソース間関連情報記憶部116から検索されて返信されると、この情報をリソース間関連情報記憶部119に格納する。 10

## 【0058】

リソース間の関連情報の収集は、アプリケーション監視エージェント241に対しても同様に行う。アプリケーション監視エージェント241の構成情報収集機能127は、予め設定されたスケジューリング設定で定期的に起動されるか、SAN性能管理ソフトウェア107の要求に基づいて起動され、自エージェント241が担当するアプリケーション240の構成情報取得機能126に対しリソース間関連情報の送信を要求し、要求した情報がSAN性能管理ソフトウェア107に返信されると、SAN性能管理ソフトウェア107はリソース間関連情報記憶部128に格納する。 20

## 【0059】

SAN性能管理ソフトウェア107の構成情報収集機能118は、アプリケーション監視エージェント241の構成情報応答機能129に対しても、各エージェント241が収集したリソース間関連情報の送信を要求する。アプリケーション監視エージェント241では、要求した情報がSAN性能管理ソフトウェア107から返信されると、これらの情報をリソース間関連情報記憶部128から検索されて返信されるとリソース間関連情報記憶部119に格納する。

## 【0060】

SAN性能管理ソフトウェア107の性能分析表示機能143は、性能管理クライアント144からの要求に応答し、メトリクス値テーブル113中のメトリック値を検索して返信する。性能管理クライアント144からの性能分析の要求に応えるため、性能分析表示機能143はネットワーク構成要素間の関係を利用する場合があるが、これに関する情報はリソース間関連情報記憶部119から検索される。 30

## 【0061】

アプリケーション監視エージェント241によるアプリケーション240の性能劣化期間の検出および格納は以下のように行われる。

## 【0062】

アプリケーション監視エージェント241のアプリケーション性能劣化期間検出機能131は、予め設定したスケジューリング設定で定期的に起動されるか、SAN性能管理ソフトウェア107のアプリケーション性能劣化情報収集機能136の要求で起動され、自エージェント241が担当するアプリケーション240のログ130を収集する。 40

## 【0063】

アプリケーション240は、性能劣化検出に必要な情報をログに書き込む。アプリケーション監視エージェント241のアプリケーション性能劣化期間検出機能127は、ログ130を読み込んで解析し、アプリケーション性能劣化期間決定ルール132に基づいて、アプリケーション128の性能劣化期間を検出し、アプリケーション性能劣化期間テーブル134に書き込む。アプリケーション性能劣化期間決定ルール132は、ストレージネットワーク管理ソフト107のアプリケーション目標性能設定機能133によって設定 50

される。アプリケーション目標性能設定機能 1 3 3 は、性能管理クライアント 1 4 4 から操作される。SAN 性能管理ソフトウェア 1 0 7 のアプリケーション性能劣化情報収集機能 1 3 6 は、予め設定されたスケジューリング設定で定期的に起動され、監視エージェント 2 0 のアプリケーション性能劣化情報応答機能 1 2 9 に対し各エージェントが検出したアプリケーション性能劣化期間の送信を要求し、要求した情報がアプリケーション性能劣化期間テーブル 1 3 4 から検索されて返信されると、アプリケーション性能劣化期間テーブル 1 3 7 に格納する。

**【 0 0 6 4 】**

SAN 性能管理ソフトウェア 1 0 7 のアプリケーション目標性能設定機能 1 3 3 の詳細は、図 3 5 を用いて詳しく説明する。アプリケーション性能劣化期間検出機能 1 3 1 の詳細は、図 3 7 を用いて詳しく説明する。

10

**【 0 0 6 5 】**

SAN 性能管理ソフトウェア 1 0 7 が収集したメトリクスの削除は以下のように行われる。

**【 0 0 6 6 】**

はじめに、SAN 性能管理ソフトウェア 1 0 7 の削除非対象リソース算出機能 1 3 8 は、アプリケーション性能劣化期間テーブル 1 3 7 を参照し、アプリケーションの性能劣化期間を得る。その後、リソース間関連記憶部 1 1 9 を参照し、該アプリケーション 2 4 0 の性能劣化期間中に、該アプリケーションと関連のあったリソースを検出する。そして、検出したリソースの結果を、削除データ制御テーブル 1 3 9 に格納する。

20

**【 0 0 6 7 】**

次に、SAN 性能管理ソフトウェア 1 0 7 のメトリクス削除機能 1 4 0 は、削除データ制御テーブル 1 3 9 に基づき、メトリクス値テーブル 1 1 3 に格納されたメトリクスを削除する。削除の際、メトリクス保存期間テーブル 1 4 1 を参照し、メトリクス保存期間を得る。メトリクス保存期間テーブル 1 4 1 の値はメトリクス保存期間設定機能 1 4 2 により設定される。メトリクス保存期間設定機能 1 4 2 は、性能管理クライアント 1 4 4 から操作される。

**【 0 0 6 8 】**

削除非対象リソース算出機能 1 3 8 の詳細は、図 3 8、図 3 9、図 4 0 を用いて詳しく説明する。メトリクス削除機能 1 4 0 の詳細は、図 4 1、図 4 2 を用いて詳しく説明する

30

**【 0 0 6 9 】**

図 3 は、リソースならびにリソース間の性能に関する依存関係の具体例を示すソフトウェア及びハードウェアの説明図である。ストレージネットワークを構成する具体的なハードウェア装置やソフトウェアごとにさまざまな種類のリソースがある。個々のストレージネットワーク中のリソースは互いに性能上の影響を及ぼし合う関係にある。

**【 0 0 7 0 】**

図 3 において、ホストサーバ A は図 1 のホストサーバ 2 0 5 であり、ホストサーバ B は図 1 のホストサーバ 2 0 6 であり、これらホストサーバ A、B は、SAN 2 を構成する SAN スイッチ 3 2 0 ~ 3 2 3 を経由してストレージサブシステム A (図 1 のストレージサブシステム 2 2 4) にアクセスする例を示す。

40

**【 0 0 7 1 】**

ホストサーバ A の SAN 2 側にはポート A ( 3 1 8 ) が設けられ、SAN スイッチ 3 2 0 のポート C ( 3 2 5 )、ポート D ( 3 2 6 )、SAN スイッチ 3 2 2 のポート H ( 3 3 2 )、ポート I ( 3 3 4 ) を介して、ストレージサブシステム A のポート N ( 3 4 1 ) に接続される。また、ポート A は、SAN スイッチのポート C ( 3 2 5 )、ポート E ( 3 2 7 ) から SAN スイッチ 3 2 3 のポート J ( 3 3 6 )、ポート L ( 3 3 8 ) を経由してストレージサブシステム A のポート O ( 3 4 2 ) に接続される。

**【 0 0 7 2 】**

ホストサーバ B の SAN 2 側にはポート B ( 3 1 9 ) が設けられ、SAN スイッチ 3 2

50

1のポートF(329)、ポートG(331)、SANスイッチ323のポートK(337)、ポートM(339)を介して、ストレージサブシステムAのポートP(343)に接続される。なお、図中SANスイッチ320~323のうち、ホストサーバA、B側に配置されたポートをホスト側ポートとし、ストレージサブシステム224側に配置されたポートをストレージ側ポートとする。

**【0073】**

ホストサーバA(205)では、アプリケーションA(240a)とアプリケーションB(240b)が稼働している。また、ホストサーバA(205)は、ファイルシステムA(208a)~ファイルシステムF(208f)とボリュームA(314)~ボリュームC(316)を有する。

10

**【0074】**

ボリュームA(314)は、後に説明するストレージサブシステムAの論理ボリュームA(344)に対しI/O命令を発行できるようにホストサーバAのOSにマウントした仮想ディスクである。ボリュームB(315)とボリュームC(316)についても同様に、それぞれ、ストレージサブシステムAの論理ボリュームB(345)と論理ボリュームC(346)をホストサーバAにマウントした仮想ディスクである。

**【0075】**

ホストサーバB(206)では、ホストサーバAと同様に、アプリケーションC(240c)が稼働しており、ホストサーバB(206)は、ファイルシステムG(208g)、ファイルシステムH(208h)とボリュームD(317)を持つ。

20

**【0076】**

SANスイッチA(320)はポート(324)~(327)を備える。SANスイッチAが持つポートのうち、番号(325)をポートC、番号(326)をポートD、番号(327)をポートEと呼ぶこととする。

**【0077】**

SANスイッチB(321)~SANスイッチD(323)についても同様に、SANスイッチBはポート(328)~(331)を、SANスイッチCはポート(332)~(335)を、SANスイッチDはポート(336)~(339)を持つ。番号(329、331、332、334、336、337、338、339)のポートをそれぞれ、ポートF、ポートG、ポートH、ポートI、ポートJ、ポートK、ポートL、ポートMと呼ぶこととする。

30

**【0078】**

ストレージサブシステムAは、物理ディスク350~357を有する。物理ディスク350~353をRAID構成した仮想ディスクが、RAIDグループA(348)で、物理ディスク(354)~(357)によるRAID構成された仮想ディスクがRAIDグループB(349)である。RAIDグループA(348)を、上位のサーバが使いやすい大きさにスライスし、スライスされた結果が論理ボリュームA(344)と論理ボリュームB(345)である。同様に、RAIDグループB(349)をスライスした結果が論理ボリュームC(346)と論理ボリュームD(347)である。

**【0079】**

サーバAでは、アプリケーションの性能劣化検出および構成情報監視のためのアプリケーション監視エージェント241と、ホストサーバAのハードならびにOSの性能情報を取得するためのホスト監視エージェント210が上記図1と同様に稼働しているものとする。ファイルシステムA(304)~ファイルシステムF(309)と、ボリュームA(312)~ボリュームC(314)、ポートA(316)は、ホストサーバA(205)で稼働するホスト監視エージェント210が、性能情報取得の対象とするリソースの一例である。ファイルシステムA~Hは、各ホストサーバA、BのOS209がデータの出入力管理を行う一つの単位であり、各ボリュームA~Dは、ファイルシステム上に作成されるファイルをマウントした外部記憶装置に格納する領域としてOS209で管理されるものである。

40

50

## 【 0 0 8 0 】

図 3 において、リソース間に引かれた線は、性能の依存関係を表わす。図 3 の例では、アプリケーション A ( 2 4 0 a ) はファイルシステム A ( 2 0 8 a ) ~ ファイルシステム D ( 2 0 8 d ) に対し I / O 処理を行う。この場合、アプリケーション A ( 2 4 0 a ) は、ファイルシステム A ( 2 0 8 a ) ~ ファイルシステム D ( 2 0 8 d ) に対し I / O 上の負荷をかけることとなり、アプリケーション A ( 2 4 0 a ) は、ファイルシステム A ( 2 0 8 a ) ~ ファイルシステム D ( 2 0 8 d ) に性能上の負荷について依存関係があることとなる。なお、性能の依存関係とは、アプリケーション 2 4 0 が発行する I / O 命令を処理する一連の装置およびソフトウェアがある場合に、これらの一連の装置およびソフトウェアの接続関係をいう。

10

## 【 0 0 8 1 】

図 3 において、ファイルシステム A 及びファイルシステム B とボリューム A を結ぶ線は、ファイルシステム A、B がボリューム A 上に配置される関係を表わしているが、この関係はまた、アプリケーション 2 4 0 a がファイルシステム A やファイルシステム B に対して操作を行った場合に、ボリューム A へ操作が繋がるため、性能上の負荷について依存関係があることも表わしている。ボリューム A ~ ボリューム C とホストサーバ A ポート A の間の線も同様に、性能の依存関係がある。

## 【 0 0 8 2 】

ホストサーバ B では、同様にアプリケーション監視エージェント 2 4 1 とホスト監視エージェント 2 1 0 が稼動しているものとする。ホストサーバ B のホスト監視エージェント 2 1 0 が性能情報取得対象とするリソースには、ファイルシステム G ( 2 0 8 g ) ~ ファイルシステム H ( 2 0 8 h ) と、ボリューム D ( 3 1 5 )、ポート B ( 3 1 7 ) がある。

20

## 【 0 0 8 3 】

SAN スイッチ A ~ SAN スイッチ D の性能情報を取得するため性能収集ソフトウェア。SAN スイッチ管理エージェント 2 3 1 が稼動しているものとする。この SAN スイッチ管理エージェント 2 3 1 が性能情報の取得対象とするリソースには、SAN スイッチ A のポート ( 3 2 2、3 2 3、3 2 6、3 2 7 )、スイッチ B のポート ( 3 2 4、3 2 5、3 2 8、3 2 9 )、スイッチ C のポート ( 3 2 2、3 2 3、3 2 6、3 2 7 )、スイッチ D のポート ( 3 2 4、3 2 5、3 2 8、3 2 9 ) がある。

## 【 0 0 8 4 】

ストレージサブシステム A では、ストレージサブシステム A の性能情報を取得するためストレージサブシステム管理エージェント 2 3 2 が稼動しているものとする。このストレージサブシステム管理エージェント 2 3 0 が情報取得対象とするリソースには、ポート N ( 3 4 1 ) ~ ポート P ( 3 4 3 ) と、論理ボリューム A ( 3 4 4 ) ~ 論理ボリューム D ( 3 4 7 )、RAID グループ A ( 3 4 8 )、RAID グループ B ( 3 4 9 )、物理ディスク ( 3 5 0 ~ 3 5 7 ) がある。

30

## 【 0 0 8 5 】

各論理ボリュームは予め RAID グループに割り当てられ、また、RAID グループは物理ディスクに割り当てられるため、これらリソース間には性能の依存関係がある。また、ホストサーバ A、B のボリュームと、同ボリュームが割り当てられるストレージサブシステムの論理ボリュームの対が決まると、この両者の間でやり取りされる入出力データの転送経路として、ホストバスアダプタ ( ホストサーバ A、B ) のポート ( A または B ) から、SAN スイッチの各ポートを経て、ストレージサブシステムのポートまでが決まる。従って、ホストサーバ A、B のボリュームにかかる入出力の負荷は経路上のポートに対する通信の負荷となるため、ボリュームおよび論理ボリュームの対と、経路上のポートの間には性能の依存関係がある。

40

## 【 0 0 8 6 】

図 3 の例では、ホストサーバ A のボリューム A は論理ボリューム A に割り当てられ、論理ボリューム A は RAID グループ A に割り当てられ、RAID グループ A は物理ディスク 3 5 0 ~ 3 5 7 に割り当てられ、ボリューム A と論理ボリューム A の対にポート A から

50

ポートC、ポートD、ポートH、ポートI、ポートNまでの経路が対応し、これらリソース間に性能の依存関係がある。

【0087】

アプリケーション240a~240cに近い側のリソースを性能の依存関係の上流と呼び、物理ディスクに近い側のリソースを性能の依存関係の下流と呼ぶこととする。図3の例では、アプリケーションAはファイルシステムAに対して上流にあたり、ファイルシステムAはボリュームAに対して上流にあたる。

【0088】

図4(A)のテーブルは、装置監視エージェント21(ホスト監視エージェント210、SANスイッチ監視エージェント231、ストレージサブシステム監視エージェント232の各々)が持つメトリクス値テーブル110の構造の一例を示す図である。 10

【0089】

メトリクス値テーブル110は、リソース識別子格納欄411とメトリクス識別子格納欄412とメトリクス値格納欄413と、T1格納欄414と、T2格納欄415からなる。

【0090】

リソース識別子格納欄411には、装置監視エージェント20が収集対象とするリソースの識別子が格納され、メトリクス識別子格納欄412には、収集するメトリクスの種別を示す識別子が格納される。装置監視エージェント20がメトリクス値格納欄413に保存するメトリクス値は、T1格納欄414で指定された時刻を開始時刻とし、T2格納欄415で指定された時刻を終了時刻とする期間内に、リソース識別子格納欄411で指定されたリソースから取得した、メトリクス識別子格納欄412で指定されたメトリクスの値が装置監視エージェント20により設定される。 20

【0091】

装置監視エージェント20が監視するメトリクス種別には、レスポンスタイム、スループット、リソース使用量、リソース使用率の4カテゴリがあり、毎秒I/O回数は、スループットの一例である。本実施例では、毎秒I/O回数をメトリクス識別子としてあげているが、装置やソフトウェアの性能を表わす他のメトリクス識別子に対する本実施例の適用を制限するものではない。

【0092】

図4(B)のテーブルは、本実施形態における装置監視エージェント20の一例である、ホストサーバA(205)を監視するホスト監視エージェントA(図1の210)が持つメトリクス値テーブル401の内容を示している。 30

【0093】

メトリクス値テーブル401は、ホスト監視エージェントAが、該ホスト監視エージェントの監視対象であるファイルシステムAや、ボリュームA、ポートA等から1時間おきにメトリクスを収集・保存していることを示している。

【0094】

メトリクス値テーブル401の第1行目には、2000年1月11日10時から2000年1月11日11時の間に、ファイルシステムAから収集した毎秒I/O回数の値が1,214.5であったことが記録されている。 40

【0095】

図5、図6、図7に示すメトリクス値テーブル501、601、701は、ホスト監視エージェントAが持つメトリクス値テーブル401と同様に、ホストサーバB(206)のホスト監視エージェントBが持つメトリクス値テーブル、ストレージ監視エージェント232が持つメトリクス値テーブル、SANスイッチ監視エージェント231が持つメトリクス値テーブルをそれぞれ示している。各装置監視エージェント20が持つメトリクス値テーブルの構造は全て共通であり、リソースの種別を格納するリソース識別子格納欄(502、602、702)、メトリクスの種別を示す値を格納するメトリクス識別子格納欄(503、603、703)、性能情報に応じたメトリクス値を格納するメトリクス値 50

格納欄(504、604、704)、開始日時を格納するT1格納欄(505、605、705)、終了日時を格納するT2格納欄(506、606、706)からなる。各格納欄の内容も同様である。

#### 【0096】

図8は、SAN性能管理ソフトウェア107が使用するメトリクス値テーブル113の構造の一例を示す図である。このテーブルの内容は、性能情報収集機能112を使って、すべての装置監視エージェント20から各々が持つメトリクス値テーブルの内容を集めて作成される。メトリクス値テーブル113は、メトリクス値テーブル110と共通の構造を持っており、リソース識別子格納欄(801)、メトリクス識別子格納欄(802)、メトリクス値格納欄(803)、T1格納欄(804)、T2格納欄(805)からなる。SAN性能管理ソフトウェア107が使用するメトリクス値テーブル113が格納する情報は、性能情報収集機能112による収集の時間遅れの期間を除けば、全ての装置監視エージェントが持つアプリケーション・ファイルシステム間関連テーブルの全ての行を合わせたものと一致する。具体的には、メトリクス値テーブル401、501、601、701の全行をマージしたものとなる。

10

#### 【0097】

図9は、グラフ形式による性能情報の表示画面の一例を示す図であり、SAN性能管理ソフトウェア107のメトリクス値テーブル113が持つ情報に基づいて、メトリクス値の時間的な推移を表示する。本画面は、SAN性能管理ソフトウェア107の性能分析表示機能143によって性能管理クライアント144に表示される。グラフの横軸(904)と縦軸(905)はそれぞれ時刻とメトリクス「毎秒I/O回数」(902)の値であり、表示内容は、複数のリソース(903)のメトリクス値の時間的な推移を示している。

20

#### 【0098】

図10は、アプリケーション監視エージェント241が持つリソース間関連情報記憶部128にはアプリケーション・ファイルシステム間関連テーブル1001が含まれることを示している。リソース間関連情報記憶部128は、リソース間の性能情報の依存関係を記憶する1つまたは複数のテーブルを格納するための記憶部である。アプリケーション・ファイルシステム間関連テーブル1001については図14にて詳しく説明する。

#### 【0099】

図11は、図10と同様に、装置監視エージェントの一つであるホスト監視エージェント210が持つリソース間関連情報記憶部116にはファイルシステム・ボリューム間関連テーブル1101と、ボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブル1102が含まれることを示している。ファイルシステム・ボリューム間関連テーブル1101については図18を用いて詳しく説明する。また、ストレージサブシステムに関連するボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブル1102については図21を用いて詳しく説明する。

30

#### 【0100】

図12は、図10と同様に、装置監視エージェント20の一つであるSANスイッチ監視エージェント231が持つリソース間関連情報記憶部116にはポート間通信経路テーブル1201が含まれることを示している。ポート間通信経路テーブル1201については図24を用いて詳しく説明する。

40

#### 【0101】

図13は、SAN性能管理ソフトウェア107が持つリソース間関連情報記憶部119にはアプリケーション・ファイルシステム間関連テーブル1301とファイルシステム・ボリューム間関連テーブル1302と、ボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブル1303とポート間通信経路テーブル1304が含まれることを示している。SAN性能管理ソフトウェア107が持つリソース間関連情報記憶部119は、全ての装置監視エージェント20が収集したリソース間関連情報を収集し、蓄えるための記憶部であり、リソース間関連情報記憶部119に含まれるテーブルの内容は、各監

50



視エージェントが持つリソース間関連情報記憶部が持つ、対応したテーブル内の行の合計となる。なお、アプリケーション・ファイルシステム間関連テーブル1301については図17を用いて詳しく説明する。

【0102】

ファイルシステム・ボリューム間関連テーブル1302については図20を用いて詳しく説明する。ボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブル1303については図23を用いて詳しく説明する。ポート間通信経路テーブル1304については図25を用いて詳しく説明する。

【0103】

図14(A)のテーブルは、アプリケーション監視エージェント241の各々が持つリソース間関連情報記憶部128に格納され、アプリケーション・ファイルシステム間関連テーブル1001の構造の一例を示す図である。

10

【0104】

アプリケーション・ファイルシステム間関連テーブル1001は、図3で説明したアプリケーションとファイルシステムとの間の性能の依存関係を記録するためのものであり、アプリケーションの識別子を格納するアプリケーション識別子格納欄1411、ファイルシステムの識別子を格納するファイルシステム識別子格納欄1412、有効期間の開始日時を格納する有効期間開始時刻格納欄1413、有効期間の終了日時を格納する有効期間終了時刻格納欄1414からなる。テーブルの各行は、依存関係があったアプリケーションとファイルシステムの組と、その組に依存関係があった期間1つに対応する。

20

【0105】

アプリケーション監視エージェント241および装置監視エージェント20が持つ構成情報収集機能は、予め設定したスケジューリング設定に従い定期的にタイマーによって起動されるか、SAN性能管理ソフトウェア107の要求で起動されると、監視対象のアプリケーション240および装置が持つ構成情報取得機能に対し、該アプリケーションおよび装置に関する性能の依存関係の送信を要求する。例えば、図3に示すホストサーバAのアプリケーションAが監視対象であれば、アプリケーションAが利用するファイルシステム一覧の送信を要求する。アプリケーションAから受信した性能の依存関係が、前回の収集時に得た関係と変化していなければ、アプリケーション監視エージェント241は、有効期間終了時刻を現在時刻で更新する。前回の収集時から性能の依存関係が変化していれば、新たな性能の依存関係を意味する行を追加する。追加する行の有効期間開始時刻は前回の収集時刻とし、有効期間終了時刻は今回の収集時刻とする。初回の収集時の場合、「前回の収集時刻」が存在しないため、その場合は、追加する行の有効期間開始時刻を過去の任意の時刻としてよい。

30

【0106】

図14(B)のテーブルは、本実施例におけるアプリケーション監視エージェントの一例である、ホストサーバA(205)上で稼働するアプリケーションA(240a)を監視するアプリケーション監視エージェントA(241)が持つリソース間関連情報記憶部に格納される、アプリケーション・ファイルシステム間関連テーブル1401の内容を示している。アプリケーション・ファイルシステム間関連テーブル1401は、テーブル1001と共通の構造をしており、アプリケーション識別子格納欄1402、ファイルシステム識別子格納欄1403、有効期間開始時刻格納欄1404、有効期間終了時刻格納欄1405からなる。アプリケーション・ファイルシステム間関連テーブル1401の第1行目には、2000年1月01日00:00~2000年1月14日13:00の間に、アプリケーションAがファイルシステムAと性能の依存関係があったことが記録されている。

40

【0107】

図15と図16に示したアプリケーション・ファイルシステム間関連テーブル1501、1602は、テーブル1401と同様に、ホストサーバA(205)上で稼働するアプリケーションB(240b)を監視するアプリケーション監視エージェントBと、ホスト

50

サーバ B ( 2 0 6 ) 上で稼働するアプリケーション C ( 2 4 0 c ) を監視するアプリケーション監視エージェント C が各々持つリソース間関連情報記憶部 1 2 8 に格納される、アプリケーション・ファイルシステム間関連テーブルの例である。

【 0 1 0 8 】

このテーブル 1 5 0 1、1 6 0 1 の構造は、アプリケーション・ファイルシステム間関連テーブル 1 4 0 1 と同様にアプリケーション識別子格納欄 ( 1 5 0 2、1 6 0 2 )、ファイルシステム識別子格納欄 ( 1 5 0 3、1 6 0 3 )、有効期間開始時刻格納欄 ( 1 5 0 4、1 6 0 4 )、有効期間終了時刻格納欄 ( 1 5 0 5、1 6 0 5 ) からなる。各格納欄の内容も同様である。

【 0 1 0 9 】

図 1 7 のアプリケーション・ファイルシステム間関連テーブル 1 3 0 1 は、SAN 性能管理ソフトウェア 1 0 7 が持つリソース間関連情報記憶部 1 1 9 が格納するアプリケーション・ファイルシステム間関連テーブルの構造の一例であり、全てのアプリケーション監視エージェントから収集したアプリケーション・ファイルシステム間の依存関係を格納する。該テーブル 1 3 0 1 の構造は、アプリケーション・ファイルシステム間関連テーブル 1 4 0 1 と同様にアプリケーション識別子格納欄 1 7 0 1、ファイルシステム識別子格納欄 1 7 0 2、有効期間開始時刻格納欄 1 7 0 3、有効期間終了時刻格納欄 1 7 0 4 からなる。該テーブル 1 3 0 1 が格納する情報は、構成情報収集機能 1 1 8 による収集の時間遅れの期間を除けば、全アプリケーション監視エージェントが持つアプリケーション・ファイルシステム間関連テーブルの全行をマージしたものと一致する。具体的には、アプリケーション・ファイルシステム間関連テーブル 1 4 0 1、1 5 0 1、1 6 0 1 の全行をマージしたものとなる。

【 0 1 1 0 】

図 1 8 ( A ) のテーブルは、ホスト監視エージェント 2 1 0 が持つリソース間関連情報記憶部に格納される、ファイルシステム・ボリューム間関連テーブル 1 1 0 1 の構造の一例である。ファイルシステム・ボリューム間関連テーブル 1 1 0 1 は、図 3 で説明したファイルシステムとボリュームとの間の性能の依存関係を記録するためのものであり、ファイルシステム識別子格納欄 1 8 1 1 とボリューム識別子格納欄 1 8 1 2、有効期間開始時刻格納欄 1 8 1 3、有効期間終了時刻格納欄 1 8 1 4 からなる。テーブルの各行は、依存関係があったファイルシステムとボリュームの組と、その組に依存関係があった期間 1 つに対応する。ホスト監視エージェントが持つファイルシステム・ボリューム間関連テーブルの更新方法は、図 1 4 で説明した方法と同様である。

【 0 1 1 1 】

図 1 8 ( B ) のテーブルは、本実施例におけるホスト監視エージェント 2 1 0 の一例である、ホストサーバ A ( 2 0 5 ) を監視するホスト監視エージェント A が持つリソース間関連情報記憶部に格納される、ファイルシステム・ボリューム間関連テーブル 1 8 0 1 の内容を示している。ファイルシステム・ボリューム間関連テーブル 1 8 0 1 は、テーブル 1 1 0 1 と共通の構造をしており、ファイルシステム識別子格納欄 1 8 0 2 とボリューム識別子格納欄 1 8 0 3、有効期間開始時刻格納欄 1 8 0 4、有効期間終了時刻格納欄 1 8 0 5 からなる。ファイルシステム・ボリューム間関連テーブル 1 8 0 1 の第 1 行目には、2 0 0 0 年 1 月 0 1 日 0 0 : 0 0 ~ 2 0 0 0 年 1 月 1 4 日 1 3 : 0 0 の間に、ファイルシステム A がボリューム A と性能の依存関係があったことが記録されている。

【 0 1 1 2 】

図 1 9 は、テーブル 1 8 0 1 と同様に、ホストサーバ B ( 2 0 6 ) を監視するホスト監視エージェント B が持つリソース間関連情報記憶部に格納される、ファイルシステム・ボリューム間関連テーブル 1 9 0 1 の例である。ファイルシステム・ボリューム間関連テーブル 1 9 0 1 の構造は、テーブル 1 8 0 1 と同様に、ファイルシステム識別子格納欄 1 9 0 2、ボリューム識別子格納欄 1 9 0 3、有効期間開始時刻格納欄 1 9 0 4、有効期間終了時刻格納欄 1 9 0 5 からなる。各格納欄の内容も同様である。

【 0 1 1 3 】

10

20

30

40

50

図20は、SAN性能管理ソフトウェア107が持つリソース間関連情報記憶部119が格納するファイルシステム・ボリューム間関連テーブル1302の例であり、全てのホスト監視エージェントから収集したファイルシステム・ボリューム間の依存関係を格納する。ファイルシステム・ボリューム間関連テーブル1302の構造は、テーブル1801と共通の構造をしており、ファイルシステム識別子格納欄2001、ボリューム識別子格納欄2002、有効期間開始時刻格納欄2003、有効期間終了時刻格納欄2004からなる。

【0114】

ファイルシステム・ボリューム間関連テーブル1302が格納する情報は、構成情報収集機能118による収集の時間遅れの期間を除けば、全ホスト管理エージェントが持つファイルシステム・ボリューム間関連テーブルの全行をマージしたものと一致する。具体的には、ファイルシステム・ボリューム間関連テーブル1801、1901の全行をマージしたものとなる。

10

【0115】

図21(A)のテーブルは、ホスト監視エージェント210が持つリソース間関連情報記憶部に格納される、ボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブル1102の構造の一例である。

【0116】

ボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブル1102は、図3で説明したボリューム・論理ボリューム・RAIDグループ・ホスト側ポート・ストレージ側ポート間の性能の依存関係を記録するためのものであり、ボリュームの識別子を格納するボリューム識別子格納欄2111、論理ボリュームの識別子を格納する論理ボリューム識別子格納欄2112、所属するRAIDグループを格納するRAIDグループ識別子格納欄2113、SANスイッチのホスト側ポートを格納するホスト側ポート識別子格納欄2114、SANスイッチのストレージ側のポートを格納するストレージ側ポート識別子格納欄2115、有効期間開始時刻格納欄2116、有効期間終了時刻格納欄2117からなる。テーブルの各行は、依存関係があったボリューム・論理ボリューム・RAIDグループ・ポートの組と、その組に依存関係があった期間1つに対応する。ホスト監視エージェントが持つボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブルの更新方法は、図14で説明した方法と同様である。

20

30

【0117】

図21(B)のテーブルは、本実施例におけるホスト監視エージェントの一例である、ホストサーバA(205)を監視するホスト監視エージェントAが持つリソース間関連情報記憶部に格納される、ボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブル2101の内容を示している。ボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブル2101は、テーブル1102と共通の構造をしており、ボリューム識別子格納欄2102、論理ボリューム識別子格納欄2103、RAIDグループ識別子格納欄2104、ホスト側ポート識別子格納欄2105、ストレージ側ポート識別子格納欄2106、有効期間開始時刻格納欄2107、有効期間終了時刻格納欄2108からなる。ボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブル2101の第1行目には、2000年1月01日00:00~2000年1月14日13:00の間に、ボリュームAが論理ボリュームA、RAIDグループA、ポートA、ポートNと性能の依存関係があったことが記録されている。

40

【0118】

図22は、テーブル2101と同様に、ホストサーバB(206)を監視するホスト監視エージェントが持つボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブル2201の例である。テーブル2201の構造は、テーブル2101と同様に、ボリューム識別子格納欄2202、論理ボリューム識別子格納欄2203、RAIDグループ識別子格納欄2204、ホスト側ポート識別子格納欄2205、ストレージ側ポート識別子格納欄22065、有効期間開始時刻格納欄2207、有効期間終了時刻格納欄22

50

08からなる。各格納欄の内容も同様である。

【0119】

図23は、SAN性能管理ソフトウェア107が持つリソース間関連情報記憶部119が格納するボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブル1303の例であり、全てのホスト監視エージェントから収集したボリューム・論理ボリューム・RAIDグループ・ホスト側ポート・ストレージ側ポート間の依存関係を格納する。図23に示すテーブルの構造は、テーブル2101と同様に、ボリューム識別子格納欄2301、論理ボリューム識別子格納欄2302、RAIDグループ識別子格納欄2303、ホスト側ポート識別子格納欄2304、ストレージ側ポート識別子格納欄2305、有効期間開始時刻格納欄2208a、有効期間終了時刻格納欄2307からなる。

10

【0120】

図23に示したテーブルが格納する情報は、構成情報収集機能118による収集の時間遅れの期間を除けば、全ホスト監視エージェントが持つボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブルの全行をマージしたものと一致する。具体的には、ボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブル2101、2201の全行をマージしたものとなる。

【0121】

図24(A)のテーブルは、ストレージネットワーク上のSANスイッチ(320~323)を監視するSANスイッチ監視エージェント231が持つリソース間関連情報記憶部116に格納される、ポート間通信経路テーブル1201の例である。ポート間通信経路テーブル1201は、図3で説明したホスト側ポートからストレージ側ポートまでの経路上にあるポート間の性能の依存関係を記録するためのものであり、SANスイッチのホスト側ポートの識別子を格納するホスト側ポート識別子格納欄2411、同じくストレージ側ポートの識別子を格納するストレージ側ポート識別子格納欄2412、SAN2内の通信経路上のポートの識別子を複数格納する通信経路上ポート群識別子格納欄2413、有効期間の開始日時を格納する有効期間開始時刻格納欄2414、有効期間の終了日時を格納する有効期間終了時刻格納欄2415からなる。

20

【0122】

テーブルの各行は、依存関係があったホスト側ポート・ストレージ側ポート・ホスト側ポートとストレージ側ポートが指定された場合の通信経路上のポート群・左記全ポートの間に依存関係があった期間1つに対応する。SANスイッチ監視エージェント231が持つポート間通信経路テーブルの更新方法は、上記図14で説明した方法と同様である。

30

【0123】

図24(B)のテーブルは、本実施例におけるSANスイッチ監視エージェント231の一例が持つリソース間関連情報記憶部116に格納される、ポート間通信経路テーブル2401の内容を示している。ポート間通信経路テーブル2401は、テーブル1201と共通の構造をしており、ホスト側ポート識別子格納欄2402、ストレージ側ポート識別子格納欄2403、通信経路上ポート群識別子格納欄2404、有効期間開始時刻格納欄2405、有効期間終了時刻格納欄2406からなる。ボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブル2401の第1行目には、2000年1月01日00:00~2000年1月14日13:00の間に、ポートA、ポートC、ポートD、ポートH、ポートI、ポートNの間に性能の依存関係があったことが記録されている。

40

【0124】

図25は、SAN性能管理ソフトウェア107が持つリソース間関連情報記憶部119が格納するポート間通信経路テーブル1304の例であり、全てのSANスイッチ監視エージェント231から収集したホスト側ポートからストレージ側ポートまでの経路上にあるポート間の性能の依存関係を格納する。図25に示すテーブルの構造は、テーブル2401と同様に、ホスト側ポート識別子格納欄2501、ストレージ側ポート識別子格納欄2502、通信経路上ポート群識別子格納欄2503、有効期間開始時刻格納欄2504、有効期間終了時刻格納欄2505からなる。

50

## 【 0 1 2 5 】

図 2 5 に示したテーブルが格納する内容は、構成情報収集機能 1 1 8 による収集の時間遅れの期間を除けば、全 S A N スイッチ監視エージェント 2 3 1 が持つポート間通信経路テーブルの全ての行を合わせたものと一致する。本実施例では、単一の S A N スイッチ監視エージェント 2 3 1 が全 S A N スイッチを監視する構成をとっているため、ポート間通信経路テーブル 1 3 0 4 の内容は、テーブル 2 4 0 1 と等しくなる。

## 【 0 1 2 6 】

図 2 6 は、アプリケーションの性能劣化判定ルールを設定するための画面の一例であり、アプリケーション目標性能設定機能 1 3 3 により性能管理クライアント 1 4 4 に表示される。当画面はアプリケーション選択領域 2 6 0 1 と目標性能入力領域 2 6 0 2、設定ボタン 2 6 0 3 からなる。ユーザは、アプリケーション選択領域 2 6 0 2 からアプリケーションを選択し、目標性能入力領域 2 6 0 3 にて目標ターンアラウンドタイム（レスポンスタイム）を入力し、設定ボタン 2 6 0 3 を押下（あるいはクリック）する。設定ボタンが押下されると、アプリケーション目標性能設定機能 1 3 3 は、選択されたアプリケーションを監視するアプリケーション監視エージェント 2 4 1 が持つアプリケーション性能劣化期間決定ルール 1 3 2 を更新する。

## 【 0 1 2 7 】

図 2 7 のアプリケーション性能劣化期間決定ルール 1 3 2 は、アプリケーション性能劣化期間決定ルールを格納するテーブルの構造の一例である。アプリケーション性能劣化決定ルール 1 3 2 は、アプリケーションの識別子を格納するアプリケーション識別子格納欄 2 7 1 1 と、アプリケーションの目標応答時間である目標ターンアラウンド時間を格納する目標ターンアラウンドタイム格納欄 2 7 1 2 からなる。アプリケーション識別子格納欄 2 7 1 1 にはユーザが選択したアプリケーションの識別子が格納され、目標ターンアラウンドタイム格納欄 2 7 1 2 にはユーザが入力した目標ターンアラウンドタイムが格納される。

## 【 0 1 2 8 】

図 2 7 のアプリケーション性能劣化期間決定ルール 2 7 0 1 は、本実施例におけるアプリケーション監視エージェントの一例である、ホストサーバ A ( 2 0 5 ) 上で稼働するアプリケーション A ( 2 4 0 a ) を監視する、アプリケーション監視エージェント A が持つアプリケーション性能劣化期間決定ルールの内容を示している。アプリケーション性能劣化期間決定ルール 2 7 0 1、テーブル 1 3 2 と共通の構造をしており、アプリケーション識別子格納欄 2 7 0 2 と、目標ターンアラウンドタイム格納欄 2 7 0 3 からなる。アプリケーション性能劣化期間決定ルール 2 7 0 1 には、アプリケーション A ( 2 4 0 a ) の目標ターンアラウンドタイムとして 1 8 0 0 秒が指定されたことが記録されている。

## 【 0 1 2 9 】

図 2 8、図 2 9 に示すアプリケーション性能劣化期間決定ルール 2 8 0 1、2 9 0 1 は、アプリケーション B を監視するアプリケーション監視エージェント B が持つアプリケーション性能劣化期間決定ルールとアプリケーション C を監視するアプリケーション監視エージェント C が持つアプリケーション性能劣化期間決定ルールの例である。

## 【 0 1 3 0 】

アプリケーション性能劣化期間決定ルール 2 8 0 1、2 9 0 1 は、テーブル 2 7 0 1 と共通の構造を持ち、アプリケーション識別子格納欄 ( 2 8 0 1、2 9 0 1 ) と、目標ターンアラウンドタイム格納欄 ( 2 8 0 2、2 9 0 2 ) からなる。各格納欄の内容も同様である。

## 【 0 1 3 1 】

図 3 0 ( A ) のテーブルは、アプリケーション監視エージェントが持つアプリケーション性能劣化期間テーブル 1 3 4 の構造の一例であり、アプリケーション監視エージェント 2 4 1 が、監視対象のアプリケーション 2 4 0 の性能劣化期間を検出し、格納した結果を示している。図 3 0 ( A ) はアプリケーションの識別子を格納するアプリケーション識別子格納欄 3 0 1 1、性能劣化の開始日時を格納する性能劣化開始時刻格納欄 3 0 1 2、性

10

20

30

40

50

能劣化の終了日時を格納する性能劣化性能劣化終了時刻格納欄 3 0 1 3 からなる。

【 0 1 3 2 】

アプリケーション監視エージェント 2 4 1 が持つアプリケーション性能劣化期間検出機能 1 3 1 は、上記したアプリケーション性能劣化期間決定ルール 1 3 2 と、アプリケーションが出力するログ 1 3 0 を参照し、自らが監視するアプリケーションの性能劣化期間を検出する。

【 0 1 3 3 】

ここで、アプリケーション監視エージェント 2 4 1 で行われる性能劣化期間検出のフローチャートを図 3 7 を用いて詳しく説明する。本実施形態では、アプリケーションの性能劣化を検出するために、アプリケーションが出力するログの情報を用いているが、その他の方法にも、アプリケーションと通信を行い、アプリケーションから性能劣化検出に必要な情報を受け取る方法や、アプリケーション自身が性能劣化を判定し、性能劣化期間テーブルに書き込む方法等もある。本実施例は、それらの方法を制限するものではない。

【 0 1 3 4 】

図 3 0 ( B ) のテーブルは、アプリケーション A ( 2 4 0 a ) を監視するアプリケーション監視エージェント A が持つアプリケーション性能劣化期間テーブル 3 0 0 1 を示している。アプリケーション性能劣化期間テーブル 3 0 0 1 は、テーブル 1 3 4 と共通の構造を持ち、アプリケーション識別子格納欄 3 0 0 2、性能劣化開始時刻格納欄 3 0 0 3、性能劣化終了時刻 3 0 0 4 からなる。アプリケーション性能劣化期間テーブル 3 0 0 1 の一行目には、アプリケーション A が 2 0 0 0 年 0 1 月 1 1 日 1 0 : 3 0 ~ 2 0 0 0 年 0 1 月 1 1 日 1 1 : 1 0 の間に、性能劣化を起こしたことが示されている。

【 0 1 3 5 】

図 3 1、図 3 2 は、テーブル 3 0 0 1 と同様に、アプリケーション B ( 2 4 0 b )、アプリケーション C ( 2 4 0 c ) を監視するアプリケーション監視エージェント B、C が持つアプリケーション性能劣化期間テーブル 3 1 0 1、3 2 0 1 を示している。図 3 1、図 3 2 の構造はテーブル 1 3 4 と共通で、アプリケーション識別子格納欄 ( 3 1 0 2、3 2 0 2 )、性能劣化開始時刻格納欄 ( 3 1 0 3、3 2 0 3 )、性能劣化終了時刻格納欄 ( 3 1 0 4、3 2 0 4 ) からなる。各格納欄の内容も同様である。

【 0 1 3 6 】

図 3 3 は、SAN 性能管理ソフトウェア 1 0 7 が持つアプリケーション性能劣化期間テーブル 1 3 7 の例であり、アプリケーション監視エージェント A、B から収集したアプリケーション性能劣化期間を格納する。図 3 3 に示すテーブルの構造はテーブル 1 3 4 と同様で、アプリケーション識別子格納欄 ( 3 3 0 1 )、性能劣化開始時刻格納欄 ( 3 3 0 2 )、性能劣化終了時刻格納欄 ( 3 3 0 3 ) からなる。図 3 3 に示したテーブルが格納する内容は、アプリケーション性能劣化情報収集機能による収集の時間遅れの期間を除けば、全アプリケーション監視エージェントが持つアプリケーション性能劣化期間テーブルの全行をマージしたものと一致する。具体的には、アプリケーション性能劣化期間テーブル 3 1 0 1、3 2 0 1、3 3 0 1 の全行をマージしたものとなる。

【 0 1 3 7 】

図 3 4 は、SAN 性能管理ソフトウェア 1 0 7 が持つ削除データ制御テーブル 1 3 9 の一例を示している。削除データ制御テーブル 1 3 9 は、アプリケーション 2 4 0 の性能劣化の開始日時を格納する性能劣化開始時刻格納欄 3 4 0 1、同じく性能劣化の終了日時を格納する性能劣化終了時刻格納欄 3 4 0 2、アプリケーション 2 4 0 に関連するリソースの識別子を格納する性能依存リソース格納欄 3 4 0 3 から構成される。当テーブルの一行は、性能劣化開始時刻格納欄 3 4 0 1 と性能劣化終了時刻格納欄 3 4 0 2 で決定される性能劣化期間内に、性能劣化を起こしたアプリケーションが、性能依存リソース格納欄 3 4 0 3 で決定されるリソースと性能上の依存関係があったことを表わす。SAN 性能管理ソフトウェア 1 0 7 が持つ削除非対象リソース算出機能 1 3 8 が、削除データ制御テーブル 1 3 9 を更新する。削除データ制御テーブル 1 3 9 の更新方法は、図 3 8、3 9 に示すフローチャートを用いて詳しく説明する。

10

20

30

40

50

## 【0138】

図34に示す削除データ制御テーブル139の例では、SAN性能管理ソフトウェア107が持つリソース間関連情報記憶部119内の4つのテーブル{アプリケーション・ファイルシステム間関連テーブル1301、ファイルシステム・ボリューム間関連テーブル1302、ボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブル1303、ポート間通信経路テーブル1304}と、アプリケーション劣化期間テーブル137とを入力として、削除非対象リソース算出機能138が処理を行い、削除データ制御テーブルを作成した例である。ただし、図34に示す削除データ制御テーブル139の例では、アプリケーションAの性能劣化期間についての情報のみ掲載している。アプリケーションBおよびアプリケーションCについて、処理を行わないわけではない。

10

## 【0139】

図35はメトリクスの保存期間を設定するための画面の一例であり、メトリクス保存期間設定機能142により性能管理クライアント144に送信され、性能管理クライアント144の表示装置上に表示される。当画面は、通常データ保存期間入力領域3501と、重要データ保存期間入力領域3502と、設定ボタン3503からなる。重要データとは、性能劣化が起きたアプリケーションの性能劣化期間中に依存関係があったリソースから収集したメトリクスを意味する。通常データは、重要データに該当しないメトリクスである。ユーザが通常データ保存期間入力領域3501と、重要データ保存期間入力領域3502にそれぞれ保存期間を入力し設定ボタン3503を押下すると、メトリクス保存期間設定機能142は、メトリクス保存期間設定テーブル141に通常データ保存期間と重要データ保存期間を格納する。

20

## 【0140】

図36は、メトリクス保存期間設定テーブル141の一例であり、通常データおよび重要データの保存期間を格納する。メトリクス保存期間設定テーブル141は、保存期間種類格納欄3601と保存期間(時間)格納欄3602からなり、通常データ保存期間の値3603と、重要データ保存期間の値3604を格納する。

## 【0141】

図37は、アプリケーション監視エージェント241が持つアプリケーション性能劣化期間検出機能131のフローチャートである。該フローチャートは、初期化处理(3701)とループ処理(3702~3707)からなる。以下、該フローチャートについて説明する。

30

## 【0142】

まず、ステップ3701で、アプリケーション性能劣化期間決定ルール132からアプリケーションの目標ターンアラウンドタイムを取得して変数「TA」の値とし、ステップ3702に進む。

## 【0143】

ステップ3702では、ログ130から未処理の処理開始時刻と処理終了時刻の組を取得し、取得した組それぞれについてループ処理を開始する。ループ内処理の開始ステップであるステップ3703に進む。

## 【0144】

ステップ3703では、処理開始時刻を変数「T1」の値とし、ステップ3404に進む。

40

## 【0145】

ステップ3704では、処理終了時刻を変数「T2」の値とし、ステップ3405に進む。

## 【0146】

ステップ3705では、処理終了時刻をT2、T1とTAとを比較する。T2 - T1がTAより大きければ、性能劣化が起きたと判定し、ステップ3706に進む。T2 - T1がTA以下であれば、性能劣化が起きていないと判定し、ステップ3707に進む。

## 【0147】

50

ステップ 3706 では、性能劣化検出対象のアプリケーション識別子、ステップ 3703 で決定した T1、ステップ 3704 で決定した T2 からなる組をアプリケーション性能劣化期間テーブル 134 に追加する。の性能に追加する。性能劣化検出対象のアプリケーション識別子は列 3011 に、T1 は列 3012 に、T2 は列 3013 にそれぞれ格納する。格納後、ステップ 3407 に進む。

【0148】

ステップ 3707 では、ループ処理対象の行がある限り、ステップ 3702 に戻る。

【0149】

上記処理により、アプリケーション監視エージェント 241 はアプリケーション性能劣化期間テーブル 134 を作成する。

10

【0150】

図 38 は、SAN 性能管理ソフトウェア 107 で実行される削除非対象リソース算出機能 138 のフローチャートである。該フローチャートは、性能劣化を起こしたアプリケーションを起点とし、RAID グループを終点とする I/O 命令が処理される経路 (I/O パス) 上のリソース群を、性能が劣化したアプリケーションと性能の依存関係にあったリソースとして検出する処理を示す。

【0151】

以下では、アプリケーションから物理ディスクまでの I/O 命令が処理される経路を I/O パスと呼ぶこととする。また、I/O パスにおいてアプリケーションに近い側を上流と呼び、物理ディスクに近い側を下流と呼ぶこととする。図 38 のフローチャートは、初期化処理 (3801) とループ処理 (3802 ~ 3805) からなる。以下、該フローチャートについて説明する。

20

【0152】

始めに、ステップ 3801 では、削除データ制御テーブル 139 の内容をクリアする初期化を実施する。クリア後、ステップ 3802 に進む。

【0153】

ステップ 3802 では、アプリケーション性能劣化期間テーブル 137 を参照し、該アプリケーション性能劣化期間テーブル 137 内の各行についてループ処理を開始する。ループ内処理の開始ステップであるステップ 3803 に進む。

【0154】

ステップ 3803 では、性能劣化を起こしたアプリケーションと性能劣化期間中に性能の依存関係のあったリソースを検出するため、性能劣化を起こしたアプリケーションを起点とする I/O パスを検出する。

30

【0155】

I/O パス検出のため、図 39 で詳しく説明する I/O パス算出サブルーチン 3901 を呼び出す。該 I/O パス算出サブルーチンを呼び出し、該 I/O パスを検出し、ステップ 3804 に進む。

【0156】

ステップ 3804 では、ステップ 3803 で検出した I/O パス上の各リソースを、性能劣化開始期間と性能劣化終了時刻の組とともに、重複を除いて削除データ制御テーブル 139 に格納する。格納後、ステップ 3805 に進む。

40

【0157】

ステップ 3805 では、アプリケーション性能劣化期間テーブル 137 に処理対象の行がある限り、ステップ 3802 に戻る。

【0158】

上記処理により、性能劣化を起こしたアプリケーション 240 を起点とする全 I/O パス上の収集対象要素を、アプリケーション 240 と性能の依存関係のあった収集要素対象とする。

【0159】

図 39 は、アプリケーション識別子と性能劣化期間を受け取り、性能劣化期間中に、該

50



アプリケーション識別子で指定されたアプリケーションを起点とするI/Oパスを算出する、I/Oパス算出サブルーチン3901である。以下、SAN性能管理ソフトウェア107で実行されるI/Oパス算出サブルーチン3901について説明する。

【0160】

まず、最初のステップ3902では、I/Oパス算出対象のアプリケーション識別子を受け取り、識別子をAPへ代入する。演算後、ステップ3903に進む。

【0161】

ステップ3903では、I/Oパス算出対象の性能劣化期間を受け取り、ステップ3904に進む。

【0162】

ステップ3904では、アプリケーション・ファイルシステム間関連テーブル1301を走査し、ステップ3902で決定したAPと、ステップ3903で決定した性能劣化期間中に、性能の依存関係があったファイルシステムを検出する。

【0163】

具体的には、アプリケーション・ファイルシステム間関連テーブル1301の全行から、アプリケーション識別子格納欄1701の値がAPと等しく、かつ、有効期間開始時刻1703と有効期間終了時刻1704で決定される期間が、性能劣化期間の少なくとも一部を含む行を絞込み、絞込みで残った全行のファイルシステム識別子格納欄1702で指定されたファイルシステムを検出する。検出したファイルシステムをF1~Fiとする。演算後、ステップ3905に進む。

【0164】

ステップ3905では、ステップ3904で検出したファイルシステムの数だけ、リソースリストを作成する。各リソースリストは、一本のI/Oパスを表わす。各リソースリストの第一要素に、検出したファイルシステムを追加する。演算後、ステップ3906に進む。

【0165】

ステップ3906では、ファイルシステム・ボリューム間関連テーブル1302を走査し、アプリケーション240の性能劣化期間中に、ステップ3904で検出したファイルシステムF1~Fiのいずれかと性能の依存関係があったボリュームを検出し、リソースリストに追加する。具体的には、ファイルシステム・ボリューム間関連テーブル1302の全行から、ファイルシステム識別子格納欄2001の値がF1~Fiのいずれかと等しく、かつ、有効期間開始時刻2003と有効期間終了時刻2004で決定される期間が、性能劣化期間の少なくとも一部を含む行を絞込み、絞込みで残った全行のボリューム識別子格納欄2002で指定されたボリュームを検出する。検出したボリュームをV1~Vjとする。検出後、ステップ3907に進む。

【0166】

ステップ3907では、ステップ3906で検出したボリュームを、該ボリュームと対応するファイルシステムをリストの第一要素として持つリソースリストに追加する。追加後、ステップ3908に進む。

【0167】

ステップ3908では、ボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブル1303を走査し、性能劣化期間中に、ステップ3906で検出したボリュームV1~Vjのいずれかと性能の依存関係があった論理ボリューム・RAIDグループ・ホスト側ポート・ストレージ側ポートの組を検出し、リソースリストに追加する。具体的には、ボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブル1303の全行から、ボリューム識別子格納欄2301の値がV1~Vjのいずれかと等しく、かつ、有効期間開始時刻2208aと有効期間終了時刻2307で決定される期間が、性能劣化期間の少なくとも一部を含む行を絞込み、絞込みで残った全行論理ボリューム識別子格納欄2302で指定された論理ボリューム、RAIDグループ識別子格納欄2303で指定されたボリューム、ホスト側ポート格納欄2304で指定されたホスト側ポート、ス

10

20

30

40

50

ストレージ側ポート格納欄 2305 で指定されたストレージ側ポート格納欄を検出する。検出した情報が持つ、ホスト側ポートとストレージ側ポートの各組を P1 ~ Pk とする。検出後、ステップ 3909 に進む。

**【0168】**

ステップ 3909 では、ステップ 3908 で検出した論理ボリューム・RAID グループ・ホスト側ポート・ストレージ側ポートの組を、該組と対応する論理ボリュームをリストの第二要素として持つリソースリストに追加する。追加後、ステップ 3910 に進む。

**【0169】**

ステップ 3910 では、ポート間通信経路テーブル 1304 を走査し、性能劣化期間中に、ステップ 3908 で検出したホスト側ポートとストレージ側ポートの組 P1 ~ Pk 10  
いずれかと性能の依存関係があった通信経路を検出し、リソースリストに追加する。具体的には、ポート間通信経路テーブル 1304 の全行から、ホスト側ポート識別子格納欄 2501 で指定されたホスト側ポートとストレージ側ポート識別子格納欄 2502 で指定されたストレージ側ポートの組が、P1 ~ Pk のいずれかと等しく、かつ、有効期間開始時刻 2504 と有効期間終了時刻 2505 で決定される期間が、性能劣化期間の少なくとも一部を含む行を絞り込み、絞り込みで残った全行の通信経路上ポート群識別子格納欄 2503 で指定されたポート群を検出する。検出後、ステップ 3911 に進む。

**【0170】**

ステップ 3911 では、ステップ 3910 で検出した通信経路上ポート群を、該ポート群と対応するホスト側ポートとストレージ側ポートを要素に含むリソースリストに追加する。ここで、該ポート群を、ホスト側ポートとストレージ側ポートの間に追加する。追加後、ステップ 3912 に進む。 20

**【0171】**

ステップ 3912 では、全てのリソースリストを返却する。

**【0172】**

本実施形態では、I/Oパス算出サブルーチン 3901 が作成するリソースリストの要素順序を、I/Oパスの上流から下流に向かうようにした。しかしながら、下流から上流に向かうようにもでき、本実施例はそのような場合を制限するものではない。また、本実施例では、I/Oパスを、アプリケーションが用いるファイルシステムを第一要素とするリソースリストによって実現しているが、さまざまな実現手法がある。本実施形態は、その 30  
のような場合を制限するものではない。

**【0173】**

図 40 は、SAN 性能管理ソフトウェア 107 が持つ削除非対象リソース算出機能 138 の他の例を示すフローチャートである。該フローチャートは、性能劣化を起こしたアプリケーションを起点とし、RAID グループを終点とする I/Oパス上のリソース群を検出し、検出した各リソースを I/Oパスの構成リソースとする全 I/Oパスを検出し、検出した I/Oパスを上流および下流に辿ることで得たリソースを性能劣化アプリケーションと性能の依存関係にあったリソースに含める処理である。該フローチャートは、初期化処理 (4001) とメインループ処理 (4002 ~ 4011) からなる。メインループ処理の内部には、2つのループ (4004 ~ 4006) と (4007 ~ 4009) がある。 40  
以下、該フローチャートについて説明する。

**【0174】**

始めに、ステップ 4001 では、削除データ制御テーブル 139 の内容をクリアする。クリア後、ステップ 4002 に進む。

**【0175】**

ステップ 4002 では、アプリケーション性能劣化期間テーブル 137 を参照し、該アプリケーション性能劣化期間テーブル 137 内の各行についてループ処理を開始する。以下では、当該行が持つ性能劣化アプリケーション識別子格納欄 3301 の値を性能劣化アプリケーションとし、性能劣化開始時刻 3302 および性能劣化終了時刻 3303 からなる期間を性能劣化期間とする。ループ内処理の開始ステップであるステップ 4003 に進 50

む。

【0176】

ステップ4003では、性能劣化を起こしたアプリケーションと性能劣化期間中に性能の依存関係のあったリソースを検出するため、性能劣化を起こしたアプリケーションを起点とするI/Oパスを検出する。I/Oパス検出のため、I/Oパス算出サブルーチン3901を呼び出す。該I/Oパス算出サブルーチンを呼び出し、該I/Oパスを検出し、ステップ4004に進む。

【0177】

ステップ4004は、全てのアプリケーションについて、I/Oパスを検出するためのループ処理の開始ステップである。ステップ4004では、アプリケーション・ファイルシステム間関連テーブル1301のアプリケーション識別子格納欄から、重複を除いた全てのアプリケーション識別子を取得し、取得した全アプリケーション識別子についてループ処理を開始する。アプリケーション識別子を取得すると、ループ内の次のステップである。ステップ4005に進む。

10

【0178】

ステップ4005では、ステップ4004でループ対象としたアプリケーションと性能劣化期間中に性能の依存関係のあったリソースを検出するため、性能劣化を起こしたアプリケーションを起点とするI/Oパスを検出する。I/Oパス検出のため、I/Oパス算出サブルーチン3901を呼び出す。該I/Oパス算出サブルーチンを呼び出し、該I/Oパスを検出し、ステップ4006に進む。

20

【0179】

ステップ4006では、ステップ4005で検出したアプリケーション毎のI/Oパスを記憶する。記憶し、ステップ4007に進む。

【0180】

ステップ4007では、ループ処理対象のアプリケーションがある限り、ステップ4004に戻る。なければ、ステップ4008に進む。

【0181】

ステップ4008では、ステップ4003で得た、性能劣化を起こしたアプリケーションを起点とするI/Oパス上の全リソースについて、ループ処理を開始する。つまり、性能劣化ループ内の最初のステップであるステップ4009に移る。

30

【0182】

ステップ4009では、ステップ4006で記憶した、アプリケーション毎のI/Oパスについて、ループ処理を開始する。ループ内の最初のステップであるステップ4010に移る。

【0183】

ステップ4010では、繰り返し対象のI/Oパスがステップ4008で繰り返し対象とするリソースを含むか否かを検出する。具体的には、I/Oパスを意味するリソースリスト内に、ステップ4008で繰り返し対象とするリソースが含まれるか否かを検出する。検査し、含まれていれば、ステップ4011に進む。含まれていなければ、ステップ4013に進む。

40

【0184】

ステップ4011では、ステップ4008で繰り返し対象とするリソースを起点とし、I/Oパスを上流および下流に辿ることによってリソースを検出する。本実施例では、上流および下流の両方に辿るが、上流のみ、あるいは下流のみに辿ることもでき、本実施例は、それらの辿り方を制限するものではない。リソースを検出後、ステップ4012に進む。

【0185】

ステップ4012では、検出したリソースを記憶する。記憶後、ステップ4013に進む。

【0186】

ステップ4013では、繰り返し処理対象のI/Oパスがある限り、ステップ4009

50

に戻る。なければステップ4014に進む。

【0187】

ステップ4014では、繰り返し処理対象のリソースがある限り、ステップ4008に戻る。なければステップ4015に進む。

【0188】

ステップ4015では、性能劣化期間・ステップ4003で検出したI/Oパス上リソースの組および、性能劣化期間・ステップ4012で検出したリソースの組を、重複を除いて削除データ制御テーブル139に追加する。追加後、ステップ4016に進む。

【0189】

ステップ4016では、アプリケーション性能劣化期間テーブル137の未処理の行がある限り、ステップ4002に戻る。なければ終了する。 10

【0190】

すなわち、ステップ4002～4007のループで、性能劣化が生じたアプリケーション240に関連するI/Oパスを検出し、ステップ4008～4016のループにより検出したI/Oパスの上流と下流のリソースを検出し、削除データ制御テーブル139が作成される。

【0191】

図41は、メトリクス削除機能140が削除するデータの種類についての説明図である。図41は、時間軸を表わす矢印4101と、時間軸上の3つの点（現在時刻4102、通常データ保存期限4103、重要データ保存期限4104）と、現在時刻から通常データ保存期限までの期間である通常データ保存期間3603と、現在時刻4102から重要データ保存期限4103までの期間である重要データ保存期間3604と、各期間内においてメトリクス値に対する処理方法の概念を表わすブロック（4105～4107）からなる。 20

【0192】

ブロック4105～4107が示す通り、メトリクス削除機能140は、通常データ保存期間3603内は全てのメトリクスを保存する（4105）。そして、通常データ保存期間3603経過後で重要データ保存期間3604内である期間は、重要データに該当するメトリクスのみ保存し、通常データは削除する（4106）。そして、重要データ保存期間3604が過ぎると、全てのメトリクスを削除する（4107）。 30

【0193】

図42は、図41で説明したメトリクス削除機能140の削除方法の一例を示したフローチャートである。該フローチャートは、初期化処理（4201～4204）とループ処理（4205～4211）からなる。

【0194】

まず、ステップ4201では、現在時刻4102を取得する。取得後、4202に進む。

【0195】

ステップ4202では、メトリクス保存期間テーブル141より通常データ保存期間3603と重要データ保存期間3604を取得する。取得後、ステップ4203に進む。 40

【0196】

ステップ4203では、現在時刻4102から、通常データ保存期間3603だけ過去に遡った時刻を、通常データ保存期限4103とする。演算後、ステップ4204に進む。

【0197】

ステップ4204では、現在時刻4102から重要データ保存期間3604だけ過去に遡った時刻を、重要データ保存期限4104とする。演算後、ステップ4205に進む。

【0198】

ステップ4205では、メトリクス保存テーブル113が持つ全行についてループ処理を開始する。ループ内処理の開始ステップであるステップ4206に進む。以下、ループ 50

内処理（ステップ4206～4210）では、ループ処理対象行が持つT2の値に応じて削除処理の実行を制御する。T2の値は、メトリクスを収集・保存した時刻を意味する。

【0199】

ステップ4206では、ループ処理対象行のT2の値と通常データ保存期限4103とを比較する。T2の値が通常データ保存期限4103以上であった場合、すなわち、メトリクス保存時刻が通常データ保存期間3603内であった場合、ステップ4211に進む。逆に、T2の値が通常データ保存期限4103より小さかった場合、すなわち、メトリクス保存時刻が通常データ保存期間3603を過ぎていた場合、ステップ4207に進む。

【0200】

ステップ4207では、ループ処理対象行のT2の値と重要データ保存期限4104とを比較する。T2の値が重要データ保存期限4104以上であった場合、すなわち、メトリクス保存時刻が重要データ保存期間3604内であった場合、ステップ4209に進む。逆に、T2の値が重要データ保存期限4104より小さかった場合、すなわち、メトリクス保存時刻が重要データ保存期間3604を過ぎていた場合、ステップ4208に進む。

【0201】

ステップ4208では、メトリクス削除テーブル113から、ループ処理対象行の削除処理を実行する。削除後、ステップ4211に進む。

【0202】

ステップ4209では、当該行が重要データであるか否かの判定を行う。該判定処理では、削除データ制御テーブル139を利用して、ループ対象行に対し次の2つの検査を実施する。両検査に当てはまれば、ループ対象行を重要データであるとする。

【0203】

（検査1）当該行のリソース識別子が、削除データ制御テーブル139が持つ性能依存リソース識別子格納欄3403に存在する。

【0204】

（検査2）当該行のT1～T2で指定される範囲の少なくとも一部が、（検査1）で存在を確認した削除データ制御テーブル139の行が持つ性能劣化開始時刻格納欄3401～性能劣化終了時刻格納欄3402の範囲に含まれる。

【0205】

上記検査1、検査2を行った結果、ループ処理対象行を重要データと判定した場合は、ステップ4211に進む。逆に、重要データでないとして判定した場合は、ステップ4208へ進む。

【0206】

ステップ4211では、ループ処理対象の行がある限り、ステップ3802に戻る。

【0207】

上記処理により、メトリクス保存期間テーブル141より通常データについては、第1の保存期間（72時間）が経過したものについては削除され、重要データについては第2の保存期間（10日間）が経過するまで保持される。

【0208】

したがって、アプリケーション240に性能劣化が生じてから10日間以内であれば、重要データとして保存された性能劣化の起きたアプリケーション240が使用するI/Oパスと、このI/Oパスの上流及び下流のメトリクス値を参照できる。これにより、アプリケーション240の性能劣化の原因を、詳細なメトリクス値（性能情報）に基づいて追究することが可能となるのである。

【0209】

一方、性能劣化に無関係なI/Oパスのメトリクス値は、通常の保存期間が経過すると自動的に削除されるため、多数のホストサーバ205～207と多数のSANスイッチ及びストレージサブシステムからなるSAN2上の膨大な性能情報のうち必要なものだけを

10

20

30

40

50

保存することができるので、性能情報（メトリクス値）を格納するためのストレージサブシステムの領域が増大するのを抑制できるのである。

【0210】

なお、上記実施形態においては、ホストサーバ205～207にアプリケーション監視エージェント241とホスト監視エージェント210や、SANスイッチ監視エージェント231、ストレージサブシステム監視エージェント232を備えたが、これらエージェントを介さず、SAN性能管理ソフトウェア107が直接、監視対象の装置やソフトウェアと通信し、性能情報を取得するようにしてもよい。

【0211】

また、上記本実施形態ではSAN2経由で性能情報を取得しているが、ストレージサブシステム224～226がLAN204に接続されている場合、ストレージサブシステム監視エージェント232は、LAN204経由でストレージサブシステム224～226およびそのポート227～229の性能情報を取得してもよい。同様に、SANスイッチ監視エージェント231も、監視対象のSANスイッチ214～216から性能情報を取得するために、LAN204を経由して通信してもよい。また、本実施例では、SANスイッチ監視エージェント231とストレージサブシステム監視エージェント232は専用の性能情報収集サーバ230上で稼働しているが、どの計算機上で稼働してもよい。アプリケーション監視エージェントおよびホスト監視エージェントについても同様で、本実施例では、ホストサーバ205上で稼働しているが、他の計算機上で稼働し、通信を用いてアプリケーションの性能情報を取得してもよい。

10

20

【0212】

なお、上記図40のステップ4011において、I/Oパスの探索を上流に辿り、性能劣化を起こしたアプリケーション240を起点とする全I/Oパス上の収集対象要素を検出し、検出した収集対象要素をI/Oパス上の装置またはソフトウェアとする全アプリケーションを検出する。そして各収集対象要素を起点として、検出した各アプリケーションのI/Oパスを上流に辿った場合の経路上の収集対象要素を、該アプリケーションと性能の依存関係のあった収集対象要素とすることができる。

【0213】

また、上記図40のステップ4011において、I/Oパスの探索を下流に辿り、性能劣化を起こしたアプリケーション240を起点とする全I/Oパス上の収集対象要素を検出し、検出した該収集対象要素をI/Oパス上の装置またはソフトウェアとする全アプリケーションを検出する。収集対象要素を起点として、検出した各アプリケーションのI/Oパスを下流に辿った場合の経路上の収集対象要素を、アプリケーション240と性能の依存関係のあった収集対象要素としてもよい。

30

【0214】

また、上記実施形態においては、所定の保存期間が経過したメトリクス値をSAN性能管理ソフトウェア107が削除する例を示したが、SAN性能管理ソフトウェア107が各エージェントに対して、削除するメトリクス値を指示するようにしてもよい。

【図面の簡単な説明】

【0215】

40

【図1】本発明の一実施形態を示すシステム構成図である。

【図2】同じく、本発明の性能管理ソフトウェアの機能ブロック図である。

【図3】図1の構成図のうち、代表的なホストサーバとSANスイッチ及びストレージサブシステムの通信状態を示す説明図である。

【図4】装置監視エージェントおよびストレージネットワーク性能管理ソフトが持つメトリクス値テーブルの構造の一例を示す図と、ホスト監視エージェントAが持つメトリクス値テーブルの内容の一例を示した図である。

【図5】ホスト監視エージェントBが持つメトリクス値テーブルの内容の一例を示した説明図である。

【図6】ストレージサブシステム監視エージェントが持つメトリクス値テーブルの内容の

50

一例を示した説明図である。

【図 7】SANスイッチ監視エージェントが持つメトリクス値テーブルの内容の一例を示した説明図である。

【図 8】ストレージネットワーク性能管理ソフトが持つメトリクス値テーブルの内容の一例を示した説明図である。

【図 9】グラフ形式による性能情報の表示画面の一例を示す説明図である。

【図 10】アプリケーション監視エージェントにおける、リソース間関連情報記憶部のテーブル構成の一例を示す説明図である。

【図 11】ホスト監視エージェントにおける、リソース間関連情報記憶部のテーブル構成の一例を示す説明図である。

10

【図 12】SANスイッチ監視エージェントにおける、リソース間関連情報記憶部のテーブル構成の一例を示す図である。

【図 13】ストレージネットワーク性能管理ソフトウェアにおける、リソース間関連情報記憶部のテーブル構成の一例を示す説明図である。

【図 14】(A)はアプリケーション監視エージェントおよびストレージネットワーク性能管理ソフトが持つアプリケーション・ファイルシステム間関連テーブルの構造の一例を示す説明図、(B)はアプリケーション監視エージェントAが持つアプリケーション・ファイルシステム間関連テーブルの内容の一例を示す説明図である。

【図 15】アプリケーション監視エージェントBが持つアプリケーション・ファイルシステム間関連テーブルの内容の一例を示す説明図である。

20

【図 16】アプリケーション監視エージェントCが持つアプリケーション・ファイルシステム間関連テーブルの内容の一例を示す説明図である。

【図 17】ストレージネットワーク性能管理ソフトが持つアプリケーション・ファイルシステム間関連テーブルの内容の一例を示す説明図である。

【図 18】(A)はホスト監視エージェントおよびストレージネットワーク性能管理ソフトが持つファイルシステム・ボリューム間関連テーブルの構造の一例を示す説明図、(B)はホスト監視エージェントAが持つファイルシステム・ボリューム間関連テーブルの内容の一例を示す説明図である。

【図 19】ホスト監視エージェントBが持つファイルシステム・ボリューム間関連テーブルの内容の一例を示す説明図である。

30

【図 20】ストレージネットワーク性能管理ソフトが持つファイルシステム・ボリューム間関連テーブルの内容の一例を示す説明図である。

【図 21】(A)はホスト監視エージェントおよびストレージネットワーク性能管理ソフトが持つボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブルの構造の一例を示す図、(B)はホスト監視エージェントAが持つボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブルの内容の一例を示す説明図である。

【図 22】ホスト監視エージェントBが持つボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブルの内容の一例を示す説明図である。

【図 23】ストレージネットワーク性能管理ソフトが持つボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブルの内容の一例を示す説明図である。

40

【図 24】(A)はSANスイッチ監視エージェントおよびストレージネットワーク性能管理ソフトが持つポート間通信経路テーブルの構造の一例を示す図、(B)SANスイッチ監視エージェントが持つポート間通信経路テーブルの内容の一例を示す図である。

【図 25】ストレージネットワーク性能管理ソフトが持つポート間通信経路テーブルの内容の一例を示す説明図である。

【図 26】アプリケーションの目標性能を設定するための画面の一例を示す説明図である。

【図 27】(A)はアプリケーションの目標性能を保存する、アプリケーション性能劣化期間決定ルールテーブルの構成の一例を示す図、(B)はアプリケーション監視エージェントAが持つアプリケーション性能劣化期間決定ルールの内容の一例を示す図である。

50

【図 28】アプリケーション監視エージェント B が持つアプリケーション性能劣化期間決定ルールの内容の一例を示す説明図である。

【図 29】アプリケーション監視エージェント C が持つアプリケーション性能劣化期間決定ルールの内容の一例を示す図である。

【図 30】(A) はアプリケーション監視エージェントおよびストレージネットワーク性能管理ソフトが使用するアプリケーション性能劣化期間テーブルの構造の一例を示す説明図、(B) はアプリケーション監視エージェント A が持つアプリケーション性能劣化期間テーブルの内容の一例を示す説明図である。

【図 31】アプリケーション監視エージェント B が持つアプリケーション性能劣化期間テーブルの内容の一例を示す説明図である。

10

【図 32】アプリケーション監視エージェント C が持つアプリケーション性能劣化期間テーブルの内容の一例を示す説明図である。

【図 33】ストレージネットワーク性能管理ソフトが持つアプリケーション性能劣化期間テーブルの内容の一例を示す説明図である。

【図 34】ストレージネットワーク性能管理ソフトが持つ、メトリクスの削除を制御するためのテーブルの構造および内容の一例を示す説明図である。

【図 35】メトリクス保存期間設定のための画面の一例を示す説明図である。

【図 36】ストレージネットワーク性能管理ソフトが持つ、メトリクス保存期間を格納するためのテーブルの構造および内容の一例を示す説明図である。

【図 37】アプリケーションの性能劣化期間を決定する処理の一例を示すフローチャートである。

20

【図 38】性能劣化期間中に、性能劣化を起こしたアプリケーションと性能の依存関係があったリソースを検出し、メトリクスの削除をコントロールするためのテーブルを更新する処理を示したフローチャートの一例である。

【図 39】性能劣化期間中に、性能劣化を起こしたアプリケーションを起点とする I / O パス上リソースを検出するためのフローチャートの一例を示す図である。

【図 40】図 38 の他の例を示し、性能劣化期間中に、性能劣化を起こしたアプリケーションと性能の依存関係があったリソースを検出し、メトリクスの削除をコントロールするためのテーブルを更新する処理を示したフローチャートの一例である。

【図 41】データ保存期間毎の、保存するメトリクスの種類を表わした説明図である。

30

【図 42】メトリクスの削除処理の一例を示すフローチャートである。

【符号の説明】

【0216】

107 SAN 性能管理ソフトウェア

205 ~ 207 ホストサーバ

210 ホスト監視エージェント

320 ~ 323 SAN スイッチ

224 ~ 226 ストレージサブシステム

230 性能情報収集サーバ

231 SAN スイッチ監視エージェント

40

232 ストレージサブシステム監視エージェント

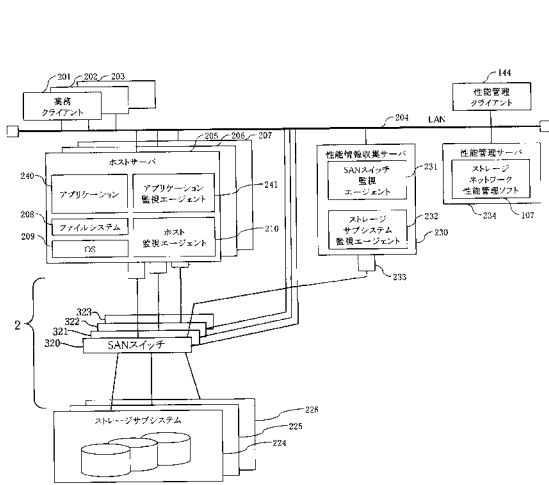
234 性能管理サーバ

240 アプリケーション

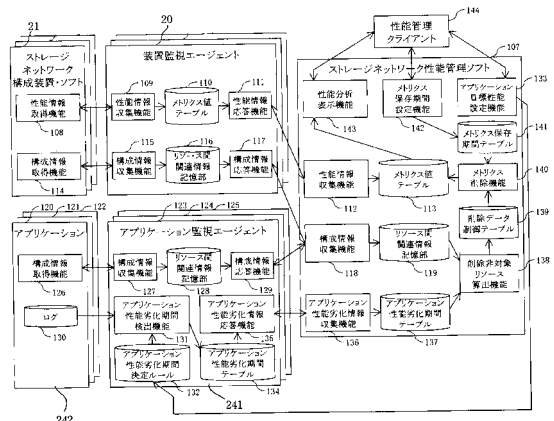
241 アプリケーション監視エージェント



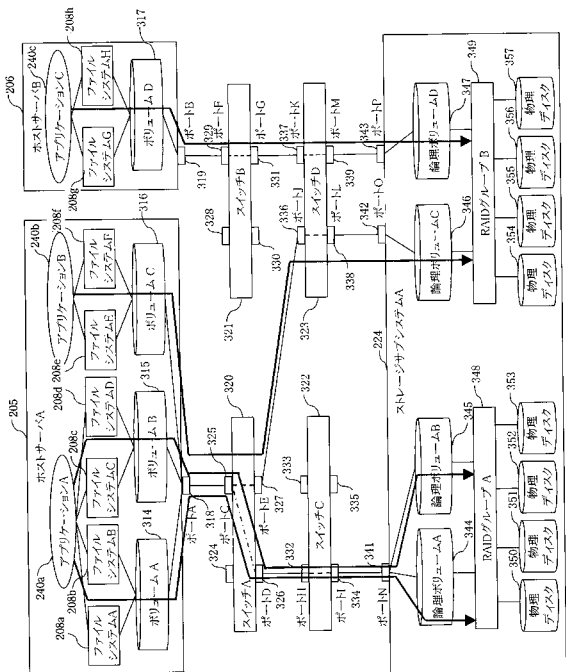
【図1】



【図2】



【図3】



【図4】

(A)

リソース識別子	外リソース識別子	外リソース値	T1	T2
ファイルシステムX	毎秒I/O回数	1,600.0	2000年01月1日 00:00	2000年01月1日 01:00

(B)

リソース識別子	外リソース識別子	外リソース値	T1	T2
ファイルシステムA	毎秒I/O回数	1,214.5	2000年01月11日 10:00	2000年01月11日 11:00
ファイルシステムA	毎秒I/O回数	503.0	2000年01月11日 11:00	2000年01月11日 12:00
ファイルシステムA	毎秒I/O回数	1,248.1	2000年01月11日 12:00	2000年01月11日 13:00
ファイルシステムA	毎秒I/O回数	1,685.4	2000年01月11日 13:00	2000年01月11日 14:00
ホリウムA	毎秒I/O回数	1,109.3	2000年01月11日 10:00	2000年01月11日 11:00
ホリウムA	毎秒I/O回数	809.0	2000年01月11日 11:00	2000年01月11日 12:00
ホリウムA	毎秒I/O回数	2,083.1	2000年01月11日 12:00	2000年01月11日 13:00
ホリウムA	毎秒I/O回数	2,103.5	2000年01月11日 13:00	2000年01月11日 14:00
ポートA	毎秒I/O回数	1,237.5	2000年01月11日 10:00	2000年01月11日 11:00
ポートA	毎秒I/O回数	3,278.3	2000年01月11日 11:00	2000年01月11日 12:00
ポートA	毎秒I/O回数	4,282.3	2000年01月11日 12:00	2000年01月11日 13:00
ポートA	毎秒I/O回数	2,435.2	2000年01月11日 13:00	2000年01月11日 14:00

【 図 5 】

501  
ホスト監視エージェントBのマトリクス値テーブル

リソース識別子	マトリクス識別子	マトリクス値	T1	T2
ファイルシステムG	毎秒I/O回数	3,223.5	2000年01月11日 10:00	2000年01月11日 11:00
ファイルシステムG	毎秒I/O回数	2,253.6	2000年01月11日 11:00	2000年01月11日 12:00
ファイルシステムG	毎秒I/O回数	892.2	2000年01月11日 12:00	2000年01月11日 13:00
ファイルシステムG	毎秒I/O回数	2192.4	2000年01月11日 13:00	2000年01月11日 14:00
...	...	...	...	...
ボリュームD	毎秒I/O回数	2,394.9	2000年01月11日 10:00	2000年01月11日 11:00
ボリュームD	毎秒I/O回数	1,403.0	2000年01月11日 11:00	2000年01月11日 12:00
ボリュームD	毎秒I/O回数	3,048.3	2000年01月11日 12:00	2000年01月11日 13:00
ボリュームD	毎秒I/O回数	1,932.4	2000年01月11日 13:00	2000年01月11日 14:00
...	...	...	...	...
ポートB	毎秒I/O回数	3,235.6	2000年01月11日 10:00	2000年01月11日 11:00
ポートB	毎秒I/O回数	2,378.0	2000年01月11日 11:00	2000年01月11日 12:00
ポートB	毎秒I/O回数	1,190.2	2000年01月11日 12:00	2000年01月11日 13:00
ポートB	毎秒I/O回数	3,394.5	2000年01月11日 13:00	2000年01月11日 14:00
...	...	...	...	...

502      503      504      505      506

【 図 6 】

601  
ストレージサブシステム監視エージェントのマトリクス値テーブル

リソース識別子	マトリクス識別子	マトリクス値	T1	T2
ポートN	毎秒I/O回数	1,492.5	2000年01月11日 10:00	2000年01月11日 11:00
ポートN	毎秒I/O回数	5,492.2	2000年01月11日 11:00	2000年01月11日 12:00
ポートN	毎秒I/O回数	5,327.3	2000年01月11日 12:00	2000年01月11日 13:00
ポートN	毎秒I/O回数	3,380.2	2000年01月11日 13:00	2000年01月11日 14:00
...	...	...	...	...
論理ボリュームA	毎秒I/O回数	721.2	2000年01月11日 10:00	2000年01月11日 11:00
論理ボリュームA	毎秒I/O回数	2,374.3	2000年01月11日 11:00	2000年01月11日 12:00
論理ボリュームA	毎秒I/O回数	3,364.1	2000年01月11日 12:00	2000年01月11日 13:00
論理ボリュームA	毎秒I/O回数	2,789.3	2000年01月11日 13:00	2000年01月11日 14:00
...	...	...	...	...
RAIDグループA	毎秒I/O回数	1,586.6	2000年01月11日 10:00	2000年01月11日 11:00
RAIDグループA	毎秒I/O回数	3,582.1	2000年01月11日 11:00	2000年01月11日 12:00
RAIDグループA	毎秒I/O回数	4,126.1	2000年01月11日 12:00	2000年01月11日 13:00
RAIDグループA	毎秒I/O回数	3,345.8	2000年01月11日 13:00	2000年01月11日 14:00
...	...	...	...	...

602      603      604      605      606

【 図 7 】

701  
SANスイッチ監視エージェントが持つマトリクス値テーブル

リソース識別子	マトリクス識別子	マトリクス値	T1	T2
ポートC	毎秒I/O回数	1,719.5	2000年01月11日 10:00	2000年01月11日 11:00
ポートC	毎秒I/O回数	6,322.3	2000年01月11日 11:00	2000年01月11日 12:00
ポートC	毎秒I/O回数	6,012.4	2000年01月11日 12:00	2000年01月11日 13:00
ポートC	毎秒I/O回数	3,581.0	2000年01月11日 13:00	2000年01月11日 14:00
...	...	...	...	...

702      703      704      705      706

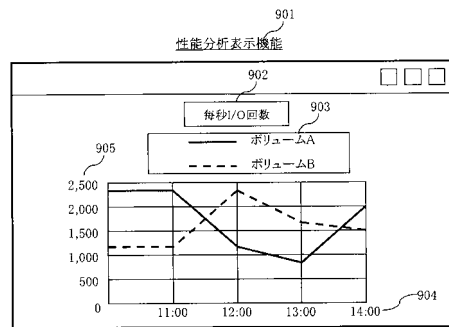
【 図 8 】

113  
マトリクス値テーブル

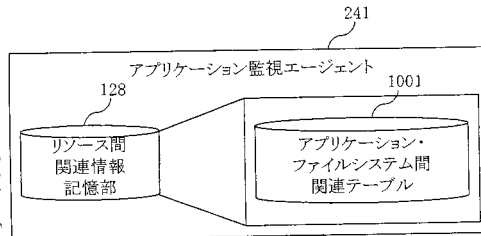
リソース識別子	マトリクス識別子	マトリクス値	T1	T2
ファイルシステムA	毎秒I/O回数	1,214.5	2000年01月11日 10:00	2000年01月11日 11:00
ファイルシステムA	毎秒I/O回数	563.0	2000年01月11日 11:00	2000年01月11日 12:00
ファイルシステムA	毎秒I/O回数	1,248.1	2000年01月11日 12:00	2000年01月11日 13:00
ファイルシステムA	毎秒I/O回数	1,685.4	2000年01月11日 13:00	2000年01月11日 14:00
...	...	...	...	...
ボリュームA	毎秒I/O回数	1,169.3	2000年01月11日 10:00	2000年01月11日 11:00
ボリュームA	毎秒I/O回数	809.0	2000年01月11日 11:00	2000年01月11日 12:00
ボリュームA	毎秒I/O回数	2,083.1	2000年01月11日 12:00	2000年01月11日 13:00
ボリュームA	毎秒I/O回数	2,103.5	2000年01月11日 13:00	2000年01月11日 14:00
...	...	...	...	...
ポートA	毎秒I/O回数	1,237.5	2000年01月11日 10:00	2000年01月11日 11:00
ポートA	毎秒I/O回数	3,278.3	2000年01月11日 11:00	2000年01月11日 12:00
ポートA	毎秒I/O回数	4,292.3	2000年01月11日 12:00	2000年01月11日 13:00
ポートA	毎秒I/O回数	2,435.2	2000年01月11日 13:00	2000年01月11日 14:00
...	...	...	...	...
ポートC	毎秒I/O回数	1,719.5	2000年01月11日 10:00	2000年01月11日 11:00
ポートC	毎秒I/O回数	6,322.3	2000年01月11日 11:00	2000年01月11日 12:00
ポートC	毎秒I/O回数	6,012.4	2000年01月11日 12:00	2000年01月11日 13:00
ポートC	毎秒I/O回数	3,581.0	2000年01月11日 13:00	2000年01月11日 14:00
...	...	...	...	...
ポートN	毎秒I/O回数	1,402.5	2000年01月11日 10:00	2000年01月11日 11:00
ポートN	毎秒I/O回数	5,492.2	2000年01月11日 11:00	2000年01月11日 12:00
ポートN	毎秒I/O回数	5,327.3	2000年01月11日 12:00	2000年01月11日 13:00
ポートN	毎秒I/O回数	3,380.2	2000年01月11日 13:00	2000年01月11日 14:00
...	...	...	...	...
論理ボリュームA	毎秒I/O回数	721.2	2000年01月11日 10:00	2000年01月11日 11:00
論理ボリュームA	毎秒I/O回数	2,374.3	2000年01月11日 11:00	2000年01月11日 12:00
論理ボリュームA	毎秒I/O回数	3,364.1	2000年01月11日 12:00	2000年01月11日 13:00
論理ボリュームA	毎秒I/O回数	2,789.3	2000年01月11日 13:00	2000年01月11日 14:00
...	...	...	...	...
RAIDグループA	毎秒I/O回数	1,586.6	2000年01月11日 10:00	2000年01月11日 11:00
RAIDグループA	毎秒I/O回数	3,582.1	2000年01月11日 11:00	2000年01月11日 12:00
RAIDグループA	毎秒I/O回数	4,126.1	2000年01月11日 12:00	2000年01月11日 13:00
RAIDグループA	毎秒I/O回数	3,345.8	2000年01月11日 13:00	2000年01月11日 14:00
...	...	...	...	...

801      802      803      804      805

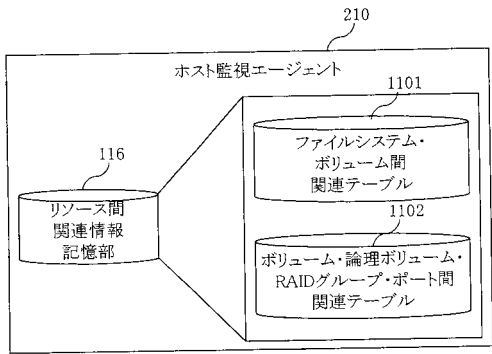
【 図 9 】



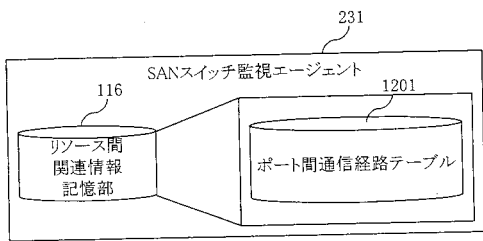
【 図 10 】



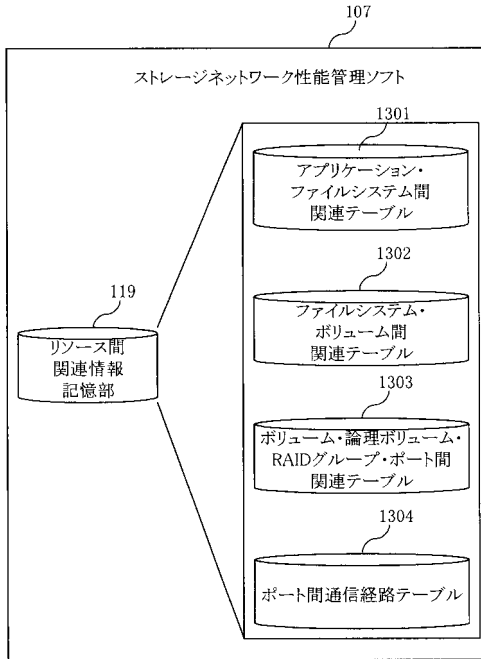
【図11】



【図12】



【図13】



【図14】

(A) Application-File System-Related Information Table (1001)

アプリケーション	ファイルシステム	有効期間開始時刻	有効期間終了時刻
アプリケーションX	ファイルシステムX	2000年01月01日00:00	2000年01月01日12:00

Labels: 1411, 1412, 1413, 1414

(B) Application Monitoring Agent A-Related Information Table (1401)

アプリケーション	ファイルシステム	有効期間開始時刻	有効期間終了時刻
アプリケーションA	ファイルシステムA	2000年01月01日00:00	2000年01月14日13:00
アプリケーションA	ファイルシステムB	2000年01月01日00:00	2000年01月14日13:00
アプリケーションA	ファイルシステムC	2000年01月01日00:00	2000年01月14日13:00
アプリケーションA	ファイルシステムD	2000年01月01日00:00	2000年01月14日13:00

Labels: 1402, 1403, 1404, 1405

【図15】

Application Monitoring Agent B-Related Information Table (1501)

アプリケーション	ファイルシステム	有効期間開始時刻	有効期間終了時刻
アプリケーションB	ファイルシステムE	2000年01月01日00:00	2000年01月14日13:00
アプリケーションB	ファイルシステムF	2000年01月01日00:00	2000年01月14日13:00

Labels: 1502, 1503, 1504, 1505

【図16】

Application Monitoring Agent C-Related Information Table (1601)

アプリケーション	ファイルシステム	有効期間開始時刻	有効期間終了時刻
アプリケーションC	ファイルシステムG	2000年01月01日00:00	2000年01月14日13:00
アプリケーションC	ファイルシステムH	2000年01月01日00:00	2000年01月14日13:00

Labels: 1602, 1603, 1604, 1605

【図17】

Application-File System-Related Information Table (1301)

アプリケーション	ファイルシステム	有効期間開始時刻	有効期間終了時刻
アプリケーションA	ファイルシステムA	2000年01月01日00:00	2000年01月14日13:00
アプリケーションA	ファイルシステムB	2000年01月01日00:00	2000年01月14日13:00
アプリケーションA	ファイルシステムC	2000年01月01日00:00	2000年01月14日13:00
アプリケーションA	ファイルシステムD	2000年01月01日00:00	2000年01月14日13:00
アプリケーションB	ファイルシステムE	2000年01月01日00:00	2000年01月14日13:00
アプリケーションB	ファイルシステムF	2000年01月01日00:00	2000年01月14日13:00
アプリケーションC	ファイルシステムG	2000年01月01日00:00	2000年01月14日13:00
アプリケーションC	ファイルシステムH	2000年01月01日00:00	2000年01月14日13:00

Labels: 1701, 1702, 1703, 1704

【 図 1 8 】

(A)

1101  
ファイルシステム・ボリューム間関連テーブル

ファイルシステム	ボリューム	有効期間開始時刻	有効期間終了時刻
ファイルシステムX	ボリュームX	2000年01月01日00:00	2000年01月01日12:00

1811 1812 1813 1814

(B)

1801  
ホスト監視エージェントAが持つ、ファイルシステム・ボリューム間関連テーブル

ファイルシステム	ボリューム	有効期間開始時刻	有効期間終了時刻
ファイルシステムA	ボリュームA	2000年01月01日00:00	2000年01月14日13:00
ファイルシステムB	ボリュームA	2000年01月01日00:00	2000年01月14日13:00
ファイルシステムC	ボリュームB	2000年01月01日00:00	2000年01月14日13:00
ファイルシステムD	ボリュームB	2000年01月01日00:00	2000年01月14日13:00
ファイルシステムE	ボリュームC	2000年01月01日00:00	2000年01月14日13:00
ファイルシステムF	ボリュームC	2000年01月01日00:00	2000年01月14日13:00

1802 1803 1804 1805

【 図 1 9 】

1901  
ホスト監視エージェントBが持つ、ファイルシステム・ボリューム間関連テーブル

ファイルシステム	ボリューム	有効期間開始時刻	有効期間終了時刻
ファイルシステムG	ボリュームD	2000年01月01日00:00	2000年01月14日13:00
ファイルシステムH	ボリュームD	2000年01月01日00:00	2000年01月14日13:00

1902 1903 1904 1905

【 図 2 1 】

(A)

1102  
ボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブル

ボリューム	論理ボリューム	RAIDグループ	ホスト側ポート	ストレージ側ポート	有効期間開始時刻	有効期間終了時刻
ボリュームX	論理ボリュームX	RAIDグループX	ポートX	ポートY	2000年01月01日00:00	2000年01月01日12:00

2111 2112 2113 2114 2115 2116 2117

(B)

2101  
ホスト監視エージェントAが持つ、ボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブル

ボリューム	論理ボリューム	RAIDグループ	ホスト側ポート	ストレージ側ポート	有効期間開始時刻	有効期間終了時刻
ボリュームA	論理ボリュームA	RAIDグループA	ポートA	ポートN	2000年01月01日00:00	2000年01月14日13:00
ボリュームB	論理ボリュームB	RAIDグループA	ポートA	ポートN	2000年01月01日00:00	2000年01月14日13:00
ボリュームC	論理ボリュームC	RAIDグループB	ポートA	ポートO	2000年01月01日00:00	2000年01月14日13:00

2102 2103 2104 2105 2106 2107 2108

【 図 2 2 】

2201  
ホスト監視エージェントBが持つ、ボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブル

ボリューム	論理ボリューム	RAIDグループ	ホスト側ポート	ストレージ側ポート	有効期間開始時刻	有効期間終了時刻
ボリュームD	論理ボリュームD	RAIDグループB	ポートB	ポートP	2000年01月01日00:00	2000年01月14日13:00

2202 2203 2204 2205 2206 2207 2208

【 図 2 0 】

1302  
ファイルシステム・ボリューム間関連テーブル

ファイルシステム	ボリューム	有効期間開始時刻	有効期間終了時刻
ファイルシステムA	ボリュームA	2000年01月01日00:00	2000年01月14日13:00
ファイルシステムB	ボリュームA	2000年01月01日00:00	2000年01月14日13:00
ファイルシステムC	ボリュームB	2000年01月01日00:00	2000年01月14日13:00
ファイルシステムD	ボリュームB	2000年01月01日00:00	2000年01月14日13:00
ファイルシステムE	ボリュームC	2000年01月01日00:00	2000年01月14日13:00
ファイルシステムF	ボリュームC	2000年01月01日00:00	2000年01月14日13:00
ファイルシステムG	ボリュームD	2000年01月01日00:00	2000年01月14日13:00
ファイルシステムH	ボリュームD	2000年01月01日00:00	2000年01月14日13:00

2001 2002 2003 2004

【 図 2 3 】

1303  
ボリューム・論理ボリューム・RAIDグループ・ポート間関連テーブル

ボリューム	論理ボリューム	RAIDグループ	ホスト側ポート	ストレージ側ポート	有効期間開始時刻	有効期間終了時刻
ボリュームA	論理ボリュームA	RAIDグループA	ポートA	ポートN	2000年01月01日00:00	2000年01月14日13:00
ボリュームB	論理ボリュームB	RAIDグループA	ポートA	ポートN	2000年01月01日00:00	2000年01月14日13:00
ボリュームC	論理ボリュームC	RAIDグループB	ポートA	ポートO	2000年01月01日00:00	2000年01月14日13:00
ボリュームD	論理ボリュームD	RAIDグループB	ポートB	ポートP	2000年01月01日00:00	2000年01月14日13:00

2301 2302 2303 2304 2305 2306 2307

【 図 2 4 】

(A)

1301  
ポート間通信経路テーブル

ホスト側ポート	ストレージ側ポート	通信経路	有効期間開始時刻	有効期間終了時刻
ポートX	ポートY	{ポートS, ポートT, ポートU, ポートV}	2000年01月01日00:00	2000年01月01日12:00

2411 2412 2413 2414 2415

(B)

2401  
SANスイッチ監視エージェントが持つ、ポート間通信経路テーブル

ホスト側ポート	ストレージ側ポート	通信経路	有効期間開始時刻	有効期間終了時刻
ポートA	ポートN	{ポートC, ポートD, ポートH, ポートI}	2000年01月01日00:00	2000年01月14日13:00
ポートA	ポートO	{ポートC, ポートE, ポートJ, ポートK}	2000年01月01日00:00	2000年01月14日13:00
ポートB	ポートP	{ポートH, ポートG, ポートK, ポートM}	2000年01月01日00:00	2000年01月14日13:00

2402 2403 2404 2405 2406

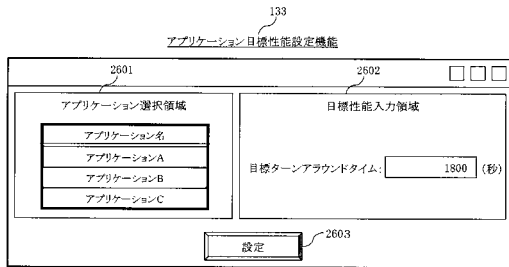
【図 25】

1304  
ポート間通信経路テーブル

ホスト側ポート	ストレージ側ポート	通信経路	有効期間開始時刻	有効期間終了時刻
ポートA	ポートM	{ポートC, ポートD, ポートH, ポートI}	2000年01月01日00:00	2000年01月14日13:00
ポートA	ポートO	{ポートC, ポートE, ポートJ, ポートK}	2000年01月01日09:00	2000年01月14日13:00
ポートB	ポートP	{ポートF, ポートG, ポートK, ポートM}	2000年01月01日00:00	2000年01月14日13:00

2501      2502      2503      2504      2505

【図 26】



【図 27】

(A) 132  
アプリケーション性能劣化期間決定ルール

アプリケーション	目標ターンアラウンドタイム(秒)
アプリケーションX	1000

2711      2712

(B) 2701  
アプリケーション監視エージェントAを持つ、アプリケーション性能劣化期間決定ルール

アプリケーション	目標ターンアラウンドタイム(秒)
アプリケーションA	1800

2702      2703

【図 28】

2801  
アプリケーション監視エージェントBを持つ、アプリケーション性能劣化期間決定ルール

アプリケーション	目標ターンアラウンドタイム(秒)
アプリケーションB	600

2802      2803

【図 29】

2901  
アプリケーション監視エージェントCを持つ、アプリケーション性能劣化期間決定ルール

アプリケーション	目標ターンアラウンドタイム(秒)
アプリケーションC	7200

2902      2903

【図 30】

(A) 134  
アプリケーション性能劣化期間テーブル

アプリケーション	性能劣化開始時刻	性能劣化終了時刻
アプリケーションX	2000年01月01日10:20	2000年01月01日10:50

3011      3012      3013

(B) 3001  
アプリケーション監視エージェントAを持つ、アプリケーション性能劣化期間テーブル

アプリケーション	性能劣化開始時刻	性能劣化終了時刻
アプリケーションA	2000年01月11日10:30	2000年01月11日11:10
アプリケーションA	2000年01月11日13:00	2000年01月11日13:40

3002      3003      3004

【図 32】

3201  
アプリケーション監視エージェントCを持つ、アプリケーション性能劣化期間テーブル

アプリケーション	性能劣化開始時刻	性能劣化終了時刻
アプリケーションC	2000年01月13日10:30	2000年01月13日13:05
アプリケーションC	2000年01月13日15:00	2000年01月13日17:55

3202      3203      3204

【図 33】

137  
アプリケーション性能劣化期間テーブル

アプリケーション	性能劣化開始時刻	性能劣化終了時刻
アプリケーションA	2000年01月11日10:30	2000年01月11日11:10
アプリケーションA	2000年01月11日13:00	2000年01月11日13:40
:	:	:
アプリケーションB	2000年01月14日01:00	2000年01月14日01:20
アプリケーションB	2000年01月14日12:00	2000年01月14日12:15
:	:	:
アプリケーションC	2000年01月13日10:30	2000年01月13日13:05
アプリケーションC	2000年01月13日15:00	2000年01月13日17:55
:	:	:

3301      3302      3303

【図 31】

3101  
アプリケーション監視エージェントBを持つ、アプリケーション性能劣化期間テーブル

アプリケーション	性能劣化開始時刻	性能劣化終了時刻
アプリケーションB	2000年01月14日01:00	2000年01月14日01:20
アプリケーションB	2000年01月14日12:00	2000年01月14日12:15

3102      3103      3104

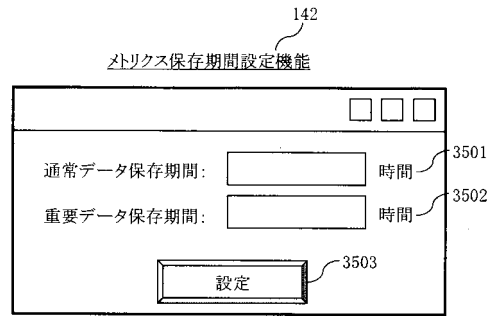
【図34】

139  
削除データ制御テーブル

性能劣化開始時刻	性能劣化終了時刻	性能依存リソース
2000年01月11日10:30	2000年01月11日11:10	ファイルシステムA
2000年01月11日10:30	2000年01月11日11:10	ファイルシステムB
2000年01月11日10:30	2000年01月11日11:10	ファイルシステムC
2000年01月11日10:30	2000年01月11日11:10	ファイルシステムD
2000年01月11日10:30	2000年01月11日11:10	ボリュームA
2000年01月11日10:30	2000年01月11日11:10	ボリュームB
2000年01月11日10:30	2000年01月11日11:10	ポートA
2000年01月11日10:30	2000年01月11日11:10	ポートC
2000年01月11日10:30	2000年01月11日11:10	ポートD
2000年01月11日10:30	2000年01月11日11:10	ポートH
2000年01月11日10:30	2000年01月11日11:10	ポートI
2000年01月11日10:30	2000年01月11日11:10	ポートN
2000年01月11日10:30	2000年01月11日11:10	論理ボリュームA
2000年01月11日10:30	2000年01月11日11:10	論理ボリュームB
2000年01月11日10:30	2000年01月11日11:10	RAIDグループA
2000年01月11日13:00	2000年01月11日13:40	ファイルシステムA
2000年01月11日13:00	2000年01月11日13:40	ファイルシステムB
2000年01月11日13:00	2000年01月11日13:40	ファイルシステムC
2000年01月11日13:00	2000年01月11日13:40	ファイルシステムD
2000年01月11日13:00	2000年01月11日13:40	ボリュームA
2000年01月11日13:00	2000年01月11日13:40	ボリュームB
2000年01月11日13:00	2000年01月11日13:40	ポートC
2000年01月11日13:00	2000年01月11日13:40	ポートD
2000年01月11日13:00	2000年01月11日13:40	ポートH
2000年01月11日13:00	2000年01月11日13:40	ポートI
2000年01月11日13:00	2000年01月11日13:40	ポートN
2000年01月11日13:00	2000年01月11日13:40	論理ボリュームA
2000年01月11日13:00	2000年01月11日13:40	論理ボリュームB
2000年01月11日13:00	2000年01月11日13:40	RAIDグループA
...	...	...

3401                      3402                      3403

【図35】



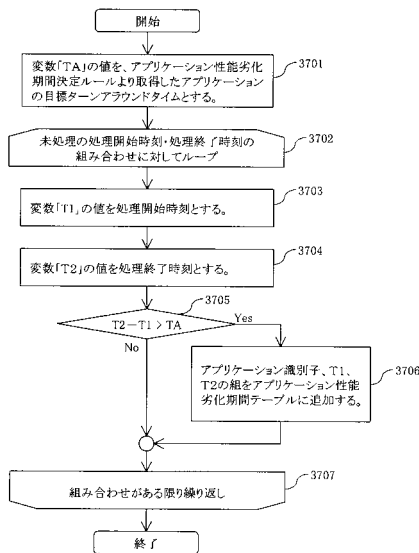
【図36】

141  
マトリクス保存期間テーブル

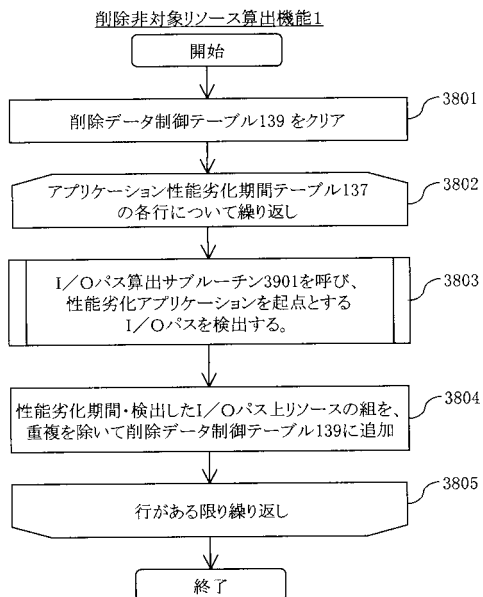
保存期間種類	保存期間(時間)
通常データ保存期間	72
重要データ保存期間	240

3601                      3602                      3603                      3604

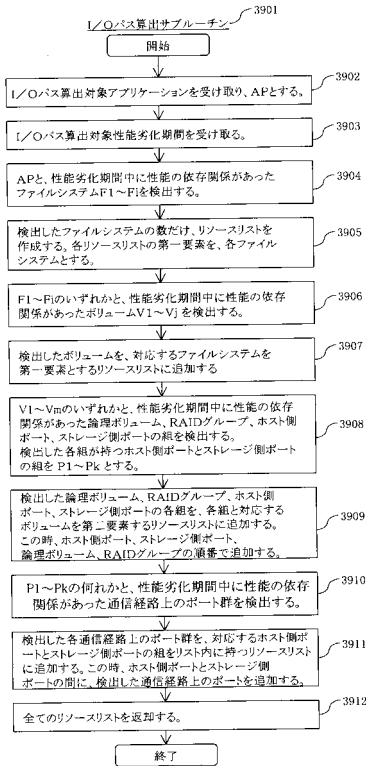
【図37】



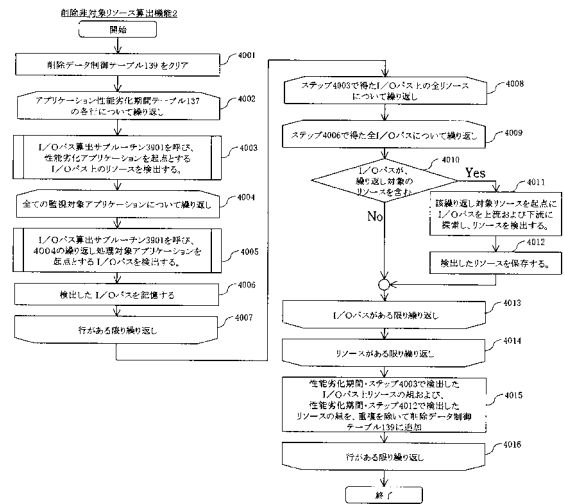
【図38】



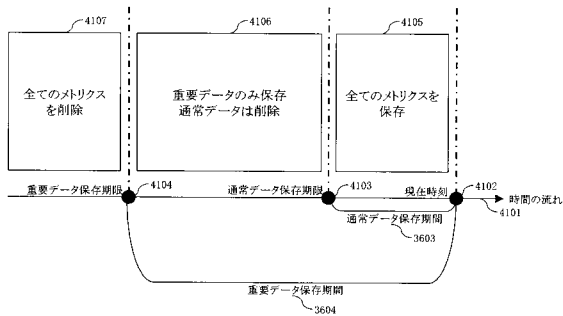
【図 39】



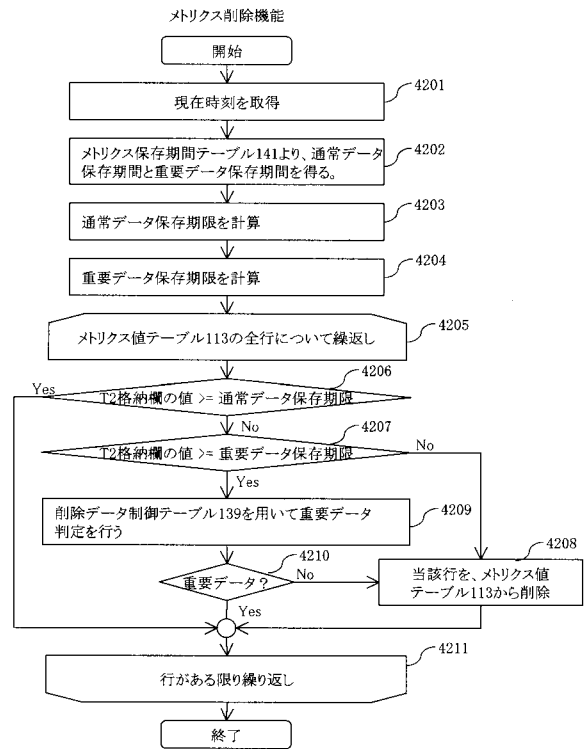
【図 40】



【図 41】



【図 42】



---

フロントページの続き

(72)発明者 大畑 秀雄

神奈川県横浜市戸塚区戸塚町5 0 3 0番地 株式会社日立製作所ソフトウェア事業部内

(72)発明者 武田 景

神奈川県横浜市戸塚区戸塚町5 0 3 0番地 株式会社日立製作所ソフトウェア事業部内

Fターム(参考) 5B014 EB04

5B065 BA01 EK05 EK07

5B082 CA14 FA04 HA08