(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2004-86914
(P2004-86914A)
(43)公開日　平成16年3月18日(2004.3.18)

(51)Int.Cl.[7]	F I			テーマコード（参考）
G06F 3/06	G06F	3/06	301F	5B065
	G06F	3/06	302Z	
	G06F	3/06	540	

審査請求 未請求　請求項の数 10　ＯＬ　（全 12 頁）

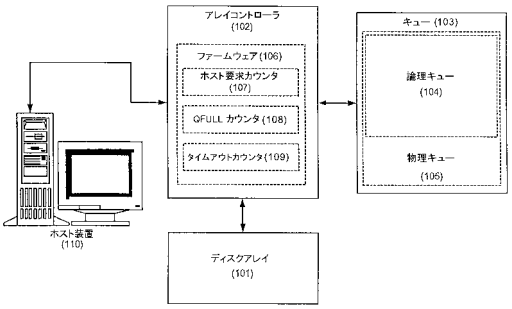
(21)出願番号　特願2003-303987(P2003-303987)
(22)出願日　平成15年8月28日(2003.8.28)
(31)優先権主張番号　10/233107
(32)優先日　平成14年8月28日(2002.8.28)
(33)優先権主張国　米国(US)

(71)出願人　503003854
ヒューレット－パッカード　デベロップメント　カンパニー　エル．ピー．
アメリカ合衆国　テキサス州　７７０７０　ヒューストン　２０５５５　ステイト　ハイウェイ　２４９
(74)代理人　100087642
弁理士　古谷　聡
(74)代理人　100076680
弁理士　溝部　孝彦
(74)代理人　100121061
弁理士　西山　清春

最終頁に続く

(54)【発明の名称】コンピュータシステム内の記憶装置のパフォーマンスの最適化

(57)【要約】
【課題】
多数のホストを接続してもホスト側タイムアウトイベントや記憶装置側のＱＦＵＬＬイベントの発生を最小限に抑える記憶装置を提供すること。
【解決手段】
データ記憶装置(101,102)は、ディスクアレイ(101)、該アレイを制御するためのアレイコントローラ(102)、及び、ホストシステムからディスクアレイ(101)へのコマンドを入れるためのキュー(103)で構成される。アレイコントローラ(102)にインストールされたプログラム(106)がキューの論理サイズ(104)を調節してパフォーマンスを最適化する。
【選択図】　図１



【特許請求の範囲】
【請求項１】
ディスクアレイ(101)と、
前記アレイ(101)を制御するためのアレイコントローラ(102)と、
ホストシステムから前記ディスクアレイ(101)へのコマンドを入れるためのキュー(103)とからなり、
前記アレイコントローラ(102)にインストールされたプログラム(106)が前記キュー(104)の論理サイズを調節してパフォーマンスを最適化する、データ記憶装置(101,102)。
【請求項２】
少なくとも１つのホスト装置(110)と少なくとも１つのデータ記憶装置(101,102)とを含むコンピュータシステム内のパフォーマンスを最適化する方法であって、前記データ記憶装置(101,102)のキューの論理サイズを調節するステップ(207,209)を含む方法。
【請求項３】
前記データ記憶装置(101,102)にさらなる記憶容量を追加するステップをさらに含む、請求項２の方法。
【請求項４】
ホスト要求をカウントするステップ(201)と、
ＱＦＵＬＬイベントをカウントするステップ(202-2)と、
タイムアウトイベントをカウントするステップ(203-2)と、
をさらに含む、請求項２の方法。
【請求項５】
発生したＱＦＵＬＬイベントの総数を受信したホスト要求の現在の数で割ることによりＱＦＵＬＬ率を計算するステップ(204)と、
前記ＱＦＵＬＬ率が所定の閾値を超えた場合、前記キューの論理サイズを増大させるステップ(207)と、
をさらに含む、請求項４の方法。
【請求項６】
発生したタイムアウトイベントの総数を受信したホスト要求の現在の数で割ることによりタイムアウト率を計算するステップ(206)と、
前記タイムアウト率が所定の閾値を超えた場合、前記キューの論理サイズを減少させるステップ(209)と、
をさらに含む、請求項４の方法。
【請求項７】
少なくとも１つのホスト装置(110)と、
少なくとも１つのデータ記憶装置(101,102)と、
前記少なくとも１つのホスト装置(110)と前記少なくとも１つのデータ記憶装置(101,102)とを接続するネットワークとからなるコンピュータシステムであって、
前記少なくとも１つのデータ記憶装置(101,102)が、
ディスクアレイ(101)と、
前記アレイ(101)を制御するためのアレイコントローラ(102)と、
ホストシステムから前記ディスクアレイへのコマンドを入れるキュー(103)とからなり、
前記アレイコントローラ(102)にインストールされたプログラム(106)が前記キューの論理サイズ(104)を調節してパフォーマンスを最適化する、コンピュータシステム。
【請求項８】
前記少なくとも１つのホスト装置(110)にインストールされたドライバ(301)と、
前記少なくとも１つのホスト装置(110)上のコマンドキュー(302)とをさらに含み、
前記ドライバ(301)が、前記少なくとも１つのデータ記憶装置(101,102)上のＱＦＵＬＬイベント(108)又はタイムアウトイベント(109)に応答して、前記ホスト装置(110)上の前記コマンドキュー(302)のサイズを調節する、請求項７のコンピュータシステム。

【請求項９】
少なくとも１つのホスト装置(110)と、少なくとも１つのデータ記憶装置(101,102)と、前記少なくとも１つのホスト装置(110)と前記少なくとも１つのデータ記憶装置(101,102)とを接続するネットワークとを含み、前記少なくとも１つのデータ記憶装置(101,102)がディスクアレイ(101)と、該アレイ(101)を制御するためのアレイコントローラ(102)と、ホストシステムから前記ディスクアレイへのコマンドを入れるためのキュー(103)とを含む、コンピュータシステムにおいてパフォーマンスを最適化する方法であって、
前記少なくとも１つのホスト装置(110)によって課せられた前記少なくとも１つのデータ記憶装置(101,102)に対する要求に応じて前記キューの論理サイズを調節するステップ(207,209)を含む方法。
【請求項１０】
前記少なくとも１つのデータ記憶装置(101,102)上のＱＦＵＬＬイベント(108)又はタイムアウトイベント(109)に応じて前記少なくとも１つのホスト装置(110)上のコマンドキューのサイズを調節するステップをさらに含む、請求項９の方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、コンピュータシステムの分野に関し、特に、ホスト装置の要求とデータ記憶装置の容量とのバランスをとる手段および方法を提供する。
【背景技術】
【０００２】
コンピュータシステムには、通常、ホストと呼ばれる１つまたは複数のコンピュータが含まれる。複数のコンピュータが使用される場合、それらのコンピュータは、データの共有を可能にするネットワークによって相互接続される。通常、そのようなネットワークは、ネットワーク接続されたそれらのコンピュータに対して追加のデータ記憶容量を提供するための１つまたは複数のデータ記憶装置も含む。一般的なデータ記憶装置はディスクアレイであり、ＲＡＩＤ（Redundant Array of Independent（またはInexpensive）Disks）と呼ばれることもある。ディスクアレイは、接続されたホストに対してデータ記憶装置を提供する２つ以上のハードディスクまたは同種のディスクである。
【０００３】
コンピュータシステムは、システム中の全コンポーネントを均等なバランスのパフォーマンスにした時に最適に動作する。ホストシステムおよび接続された記憶装置は、記憶装置のホスト要求を満たす能力がホストによって生成された作業負荷とおよそ等価になるように、通常バランスを取らなければならない。さらに複雑な構成の場合、多数のホストが１つまたは複数の記憶装置との間でデータを転送することが可能になっている場合がある。複数のホストが１つの記憶装置にアクセスする場合、その記憶装置は、接続されたすべてのホストのパフォーマンス要求に対応できることが重要である。パフォーマンスの観点から記憶装置の能力がホストの能力よりも大幅に低い場合、記憶装置はシステムパフォーマンス全体における制限要素になる可能性がある。
【０００４】
ファイバーチャネル（ＦＣ）テクノロジは、ホスト装置と記憶装置とを接続またはネットワーク化するテクノロジである。ファイバーチャネルテクノロジは、１つの記憶装置に接続された何千ものホストを含むことが可能な構成を可能にする。ファイバーチャネルは、ＳＣＳＩ（小型コンピュータシステムインタフェース）プロトコル、たとえばＳＣＳＩ－３通信プロトコルを使用する。したがって、何千に及ぶ可能性があるホストの各々は、ディスクアレイ内で最大６５，５３６までの論理記憶装置（ＬＵＮ）をアドレス指定することができる。今日では、何千ものＬＵＮを作成する能力を有するファイバーチャネルテクノロジを用いて通信するディスクアレイが出荷されている。
【０００５】
ディスクアレイ構成は、４つ以下のドライブを有するアレイから１００を超えるドライ

ブを有するアレイまで、さまざまである。最終的に、ディスクアレイのパフォーマンスは、アレイ中のディスク数によって制限される。これは、すべての作業が最終的にディスクに渡されることになるからである。

【０００６】

明らかに、ファイバーチャネルテクノロジは、１つの記憶装置に多数のホストを接続することをユーザに勧めるものである。多数のホストは、ディスクアレイ等の記憶装置に対して莫大な量の作業を命令する可能性がある。したがって、そのような環境にいるユーザは、何百ものディスクを有するエンタープライズディスクアレイを用いた作業に慣れているので、多数のホストをサポートするのに必要なパフォーマンスを受け入れることができる場合がある。

【０００７】

高要求環境では、各ホストが複数のＬＵＮに対する複数の入出力（Ｉ／Ｏ）要求を常に有する場合がある。そうした要求の各々は、処理されるまでキューに入れる必要があり、キューが一杯の場合は拒絶する必要がある。

【０００８】

システムパフォーマンスは、広範囲に研究されてきた複雑な課題である。多数のパフォーマンス要因および変形を考慮すると、ホスト／アプリケーション構成によって生成される作業負荷が記憶システムの能力を超えしまうか否かをユーザが設計の時点で知ることは、不可能ではないが非常に困難なものとなりうる。したがって、記憶パフォーマンスとのバランスがとれていないシステムをユーザが設定してしまうことは、決してめずらしいことではない。

【０００９】

１つのアンバランスなシステムにおいて、ユーザは、接続されたホストによって生成される作業負荷を大幅に超える記憶装置のパフォーマンス能力を有する場合がある。ユーザは、そのソリューション全体のパフォーマンスに満足できず、所望のパフォーマンス結果を達成するために、さらにホスト容量を追加しなければならない場合がある。

【００１０】

本発明は、ホスト／アプリケーションの作業負荷が接続された記憶装置の能力を超過する相補的なシナリオを扱う。本明細書で使用する「過剰構成（over-configuration）」という用語は、ホスト／アプリケーションのパフォーマンス要求が記憶装置のパフォーマンス能力を超過するシナリオを説明するのに用いられる。別の言い方をすると、システム設計者がソリューション全体の要求に対して記憶装置を十分に供給しなかったということである。

【００１１】

過剰構成されたシステムは、不満よりも、もっと深刻な問題を引き起こす場合がある。データ記憶アレイがシステムパフォーマンス要求を満たすことができない場合、ユーザアプリケーションの応答回数が許容できないほど多くなる。もっと極端な場合、アプリケーションエラー、サーバクラッシュおよびホストクラスタ障害などの悲惨な事態の兆候が現れる場合もある。それらの悲惨な事態が発生する理由は、ホストのソフトウェア（ドライバ、アプリケーション、ボリュームマネージャなど）が、記憶装置によって要求が完了されるのを待つ時間を制限しているからである。ディスクアレイ等の記憶装置の応答時間がホストソフトウェアによって課せられた制限時間を超えると、悲惨な事態が発生する可能性がある。明らかに、過剰構成を検出する方法および過剰構成を完全に回避する方法が必要とされている。

【００１２】

現在、過剰構成問題は、ユーザが試行錯誤によって発見している。この発見は、大抵の場合サポート要求によって行なわれ、その場合、サポート要員はシステムがアンバランスな状態で動作していることを推測できるにすぎない。これは、パフォーマンス問題が発生した際に、それらのアンバランスな構成を回避する方法をユーザに知らせることに重点をおく受動的な解決方法であり、それらのアンバランスな構成を検出および軽減する方法を

ユーザに知らせるものではない。

【００１３】

パフォーマンスの複雑性および動的性質が原因で、この発見は、非常に誤りを引き起こしがちな処理になる可能性がある。アンバランスな構成は極めて一般的である。アンバランスな構成が設定されると、ユーザはまず、システム全体のパフォーマンスが許容できないものであることを確認しなければならない。ユーザは、手がかりを提供する様々なツールを用いて、十分に実行されていないシステムを確認する。しかしながら、この処置には、構成、アプリケーション、及び、様々な情報源から取得したその他多数の情報の断片を手作業で検査することが含まれる。この処置は専門家による自学自習の分析で終わるのが常であり、専門家はそのシステムがアンバランスであると結論する。この結論が得られると、次のステップとして、さらに記憶装置を追加し、パフォーマンス問題が解決されることを期待することになる。

【００１４】

一般に、Ｉ／Ｏ要求が記憶装置に送信されると、そのＩ／Ｏ要求は、許容または拒絶される。許容された要求は、サービスキューに配置され、さらなる処理を待つ。新たな要求を取り込んでキューに入れる記憶装置の能力を超えると、その要求は拒絶され、キューが満杯であるというステータス（ＱＦＵＬＬ）とともにホストシステムに返される。ＱＦＵＬＬはこの状態についての適当な戻りステータスであるが、記憶装置によっては、ＱＦＵＬＬではなくＢＵＳＹを返す場合もある。これは、ホストシステムドライバがＢＵＳＹ状態およびＱＦＵＬＬ状態について一般に異なる再試行挙動を示すという点で重要である。

【００１５】

記憶装置はＱＦＵＬＬイベントが発生し得ないほどそのサービスキューを大きくすることもできるが、この解決方法にも限界がある。サービスキューのサイズは、実際には、ホストシステムドライバおよびアプリケーションによって課せられたタイムアウト時間によって制限される。記憶装置によりアプリケーションが永久待機させられるということが絶対に起こらないようにするため、ホストシステムドライバ及びいくつかのソフトウェアアプリケーションは、Ｉ／Ｏ要求に対してタイムアウトを課している。要求がタイムアウトすると、ホストは、記憶装置に何らかの問題があることを推定する。この状態から復帰するため、ホストシステムは、そのＩ／Ｏを中止すべきことを記憶装置に命令し、その後その要求を再試行する（何らかの有限数の再試行回数まで）。

【００１６】

キューに入るＩ／Ｏ要求の数が極めて多い場合、それらの要求を処理する際の待ち時間は、ホストシステムのドライバ及び／又はホストシステムのアプリケーションによって課せられたタイムアウト制限を超えてしまう場合がある。そうしたタイムアウトが発生する理由は、単に、記憶装置がサービスキュー中にある要求の数を許容時間内に処理することができないからである。そのような理由から、記憶装置のサービスキューのサイズについては実際上制限がある。

【００１７】

ＱＦＵＬＬイベントおよびタイムアウトの両方に対するホストの反応動作は、パフォーマンスの点で費用がかかる。実際に、両イベントはシステム内を伝搬し、実際の、すなわち知覚される障害を生じさせる可能性がある。両イベントは、ホストプログラム（すなわち、ファイバーチャネルドライバ）においてエラー事象として扱われる。それらのイベントは、ホストシステムのログに現れ、重大な障害として扱われる場合が多い。この解釈は、他の挙動次第で正しい場合もあればそうでない場合もある。

【発明の開示】

【発明が解決しようとする課題】

【００１８】

ＱＦＵＬＬイベントおよびタイムアウトイベントの処理は、記憶装置の設計者にジレンマを残す。一方、記憶サービスキューのサイズは、最大まで大きくすることが望ましい。キューを大きくすることは、より多数のホストが接続できるようになることを意味する。

これにより、不要なＱＦＵＬＬイベントを生じさせることなくアレイの接続性を向上させることができる。ホストおよびアプリケーションのＩ／Ｏ要求特性が適当なものであれば、比較的低速な記憶装置についても、大量のホストを備えたバランスのとれたシステムを構成することが可能であるかもしれない。一方、記憶装置の設計者は、最悪の場合のアクセスパターンを有する大型の構成であってもタイムアウトイベントを発生させることなく対処することができるように、キューサイズを十分小さく保つことを望んでいる。明らかに、システム設計者がこれらの競合する考慮事項のバランスをとろうとすることは、従来は極めて困難であった。

【課題を解決するための手段】

【００１９】

多数の考え得る実施形態のうちの１つにおいて、本発明は、ディスクアレイと、ディスクアレイを制御するアレイコントローラと、ホストシステムからディスクアレイへのコマンドを入れるためのキューとを有するデータ記憶装置を提供する。アレイコントローラにインストールされた例えばファームウェアなどのプログラムは、パフォーマンスを最適化するためにキューの論理サイズを調節する。

【００２０】

本発明の他の実施形態は、少なくとも１つのホスト装置と少なくとも１つのデータ記憶装置とを有するコンピュータシステム内のパフォーマンスを最適化する方法であって、データ記憶装置のキューの論理サイズを調節することを含む方法も提供する。

【００２１】

本発明の他の実施形態は、少なくとも１つのホスト装置と、少なくとも１つのデータ記憶装置と、少なくとも１つのホスト装置と少なくとも１つのデータ記憶装置とを接続するネットワークとを有するコンピュータシステムも提供する。データ記憶装置は、ディスクアレイと、ディスクアレイを制御するアレイコントローラと、ホストシステムからディスクアレイへのコマンドを入れるキューとを有する。アレイコントローラにインストールされたプログラムは、パフォーマンスを最適化するためにキューの論理サイズを調節する。

【００２２】

本発明の他の実施形態は、少なくとも１つのホスト装置と、少なくとも１つのデータ記憶装置と、少なくとも１つのホスト装置と少なくとも１つのデータ記憶装置とを接続するネットワークとを有するコンピュータシステムであって、前記少なくとも１つのデータ記憶装置がディスクアレイと、ディスクアレイを制御するアレイコントローラと、ホストシステムからディスクアレイへのコマンドを入れるキューとを有するコンピュータシステム内のパフォーマンスを最適化する方法も提供する。この方法は、少なくとも１つのホスト装置によって課せられた少なくとも１つのデータ記憶装置への要求に応じて、キューの論理サイズを調節することを含む。

【００２３】

添付の図面は、本発明の様々な実施形態を例示する。図面は、下記の説明とあわせて本発明の原理を例示および説明するものである。例示の実施形態は本発明の例であり、本発明の特許請求の範囲を限定するものではない。

【００２４】

図面を通して同一の符号は類似要素を指すものであり、必ずしも同一の要素を指すものではない。

【発明を実施するための最良の形態】

【００２５】

本発明は、ユーザおよびサポート要員がキューサイズおよび応答時間に関してシステムについて最適な選択を行うことを可能にする新たなアレイキュー手段を導入している。本発明の原理に従う記憶装置のファームウェア又は他のプログラムによって、サポート要員は、システムの動作中にアレイのサービスキューのサイズを動的に変更することができる。変更はほぼ瞬時に反映される。

【００２６】

図１は、本発明の一実施形態による、ホスト装置とデータ記憶装置とを備えた改良されたコンピュータシステムを示すブロック図である。本明細書および特許請求の範囲で使用する「記憶装置」という用語は、ディスクアレイ、ディスクドライブ、テープドライブ、任意のＳＣＳＩデバイスまたはファイバーチャネル装置を含む任意のデータ記憶装置を指すものとして使用するが、それらに限定はしない。図１に示すように、例示のシステムは、１つまたは複数のディスクアレイ（１０１）を含むデータ記憶装置に接続された１つまたは複数のホスト装置（１１０）を含む。

【００２７】

ディスクアレイ（１０１）はアレイコントローラ（１０２）によって制御される。アレイコントローラ（１０２）は、ホスト装置（１１０）からＩ／Ｏ要求を受信し、それらの要求に応じてディスクアレイ（１０１）を制御する。アレイコントローラ（１０２）は、ホスト要求を処理することができるようになるまでそれらをキュー（１０３）に入れておく。様々なホスト装置（１１０）がこの１つの共通キューに対して要求を提出する。

【００２８】

本発明の原理によると、キューの物理サイズ（１０５）はキューの論理サイズ（１０４）と区別される。キューの物理サイズ（１０５）は、ファームウェア、ソフトウェアまたは他のプログラム（以下「ファームウェア」と呼ぶ）（１０６）がアレイコントローラ（１０２）にインストールされた時点でサポートされていた最大論理キューサイズ（１０４）を収容できるように設定される。論理キューサイズ（１０４）は、キュー（１０３）がホスト（１１０）にＱＦＵＬＬイベントを返す前に格納しているホストＩ／Ｏ要求の数である。論理キューサイズ（１０４）は、システム動作中に、アレイコントローラ（１０２）のファームウェア（１０６）により、物理キューサイズ（１０５）の最大値まで動的に調節することができる。ファームウェア（１０６）のアルゴリズムは、論理キューサイズ（１０４）を物理キューサイズ（１０５）よりも増大させようとする試みを防止するように構成することが好ましい。

【００２９】

また、本発明によるディスクアレイファームウェア（１０６）は、本記憶装置システムの診断および調節に利用可能なキューパフォーマンス情報を追跡してログ記録する３つのイベントカウンタも導入している。それらのカウンタは、記憶装置が遭遇したホスト要求（１０７）、ＱＦＵＬＬイベント（１０８）およびタイムアウトイベント（１０９）の数を追跡する。実際には、記憶装置は、要求の処理の中止を通知された回数をカウントすることにより、タイムアウトイベントを間接的にカウントすることができる。中止の大半がタイムアウトの結果生じるものであると仮定した場合。

【００３０】

ユーザおよびサポートチームメンバは、調節可能な論理キューサイズ（１０４）とともに、記憶装置のそれらのカウンタ（１０７～１０９）を用いて取り込んだ情報を用いて、自分達のシステムの作業負荷・パフォーマンスを調節することができる。目的は、ＱＦＵＬＬイベントを実質的になくすのに十分な程度大きいと同時に、タイムアウトイベントを最小限にするのに十分な程度小さい論理キューサイズ（１０４）を見つけることである。ＱＦＵＬＬイベントやタイムアウトを完全になくすことは、実際的でない。良くバランスのとれた正常なコンピュータシステムであっても、ＱＦＵＬＬイベントおよびタイムアウトは、稀なイベント又は異常なイベント（作業負荷の予期せぬスパイク等）に起因して発生する場合がある。

【００３１】

目的は、より正確には、論理キューサイズ（１０４）を操作して２つの割合、すなわちホスト要求の数に対するＱＦＵＬＬイベントの数の割合（ＱＦＵＬＬ率）と、ホスト要求の数に対するタイムアウトイベントの数の割合（タイムアウト率）とを最小化することであるべきである。目的は、ＱＦＵＬＬ率とタイムアウト率との両方をゼロに近づける論理キューサイズ（１０４）を見付けることになる。

【００３２】

図２は、本発明の一実施形態による、ホスト装置およびデータ記憶装置を備えたコンピュータシステムを動作させる方法を示すフロー図である。図２に示すように、論理キューサイズを最適化するプロセスは、ＱＦＵＬＬ率およびタイムアウト率を最小化すなわちゼロに近づけようと試みる。

【００３３】

まず、コントローラファームウェアのカウンタがホスト要求をカウントする。ホスト要求を受信（２００）する毎に、ホスト要求カウンタをインクリメントする（２０１）。ホスト要求によってＱＦＵＬＬイベントが生じた場合（２０２－１）、ＱＦＵＬＬイベントカウンタをインクリメントする（２０２－２）。同様に、ホスト要求が処理されなかったためにタイムアウトイベントが生じた場合（２０３－１）、タイムアウトカウンタをインクリメントする（２０３－２）。

【００３４】

次に、ＱＦＵＬＬイベントの数を受信したホスト要求の現在の数で除算することにより、ＱＦＵＬＬ率を計算する（２０４）。そして、このＱＦＵＬＬ率を閾値と比較する（２０５）。理想的なシステムではこの閾値がゼロであるが、実際のシステムではそれよりも高く設定される。ＱＦＵＬＬ率が許容可能な閾値を超過すると（２０５）、アルゴリズムは論理キューサイズを増大させる（２０７）。このようにして追加のキュー容量を与えることにより、ＱＦＵＬＬイベントの数が減少する。

【００３５】

次に、タイムアウトイベントの数を受信したホスト要求の現在の数で除算することにより、タイムアウト率を計算する（２０６）。ここでも、理想的なシステムではこの閾値がゼロであるが、実際のシステムではそれよりも高くすることが多い。タイムアウト率が許容可能な閾値を超過すると（２０８）、アルゴリズムは論理キューサイズを減少させる（２０９）。このようにして実行待ちの要求の数を制限することにより、タイムアウトイベントが防止される。

【００３６】

両方の割合がそれぞれの閾値を超過した場合（２１０）、記憶装置は、生成されているホスト作業負荷に対処することができなくなる。これは、異常に高いホスト作業負荷に起因した一時的な状態である可能性もある。しかしながら、この状態が持続するならば、システムは恐らく過剰構成されている。そのような場合、ファームウェアは、システム管理者に対し、システムが過剰構成されている可能性があるということの通知を生成することができる。そして、システム管理者は正しい処置を考えることができる。正しい対応は、ディスクアレイの能力を増大させる（キャッシュ、アレイコントローラまたはディスクを追加することにより）ことや、単に別のディスクアレイを追加することになるであろう。

【００３７】

上記の説明では、ディスクアレイその他の記憶装置のキューに重点を置いている。しかしながら、ホスト要求はホスト自体のキューも通過する。また、ホストドライバおよびホストバスアダプタ（ＨＢＡ）も、それら自体の内部キューで要求を処理する。それらのキューも本発明の原理に基づいて動的に調節することができるので、ホスト内で使用されるアルゴリズムを理解することは重要である。

【００３８】

記憶装置では、Ｉ／Ｏ要求を管理する手段としてコマンドサービスキューが使用される。コマンドサービスキューは、低頻度、短期間の高い要求も受け入れるとともに、通常の負荷も処理できる程度に十分大きいものである必要がある。ディスクアレイのＩ／Ｏ処理速度がホストのＩ／Ｏ要求速度と同じかそれよりも速い場合、サービスキューの利用率は、長期にわたって変化しないかまたは減少してゆく。ホストのＩ／Ｏ要求速度がそれらの要求をディスクアレイが処理できる速度を超過した場合、キュー容量を使い果たすまでサービスキューレベルが増大してゆく。

【００３９】

複雑なストレージエリアネットワーク（ＳＡＮ）環境におけるディスクアレイには、複

数のホストによって要求がポストされる。それらのホストの各々は、バースト要求を送信することができる。本明細書で使用する「バースト」という用語は、要求の数が一定であること、及び、それらの要求がディスクアレイの要求を処理する能力を超える速度で伝達されることを意味する。従って、バースト要は、アレイのサービスキューの需要量を増大させる結果となる。各ホストは、自分が送信するバーストのサイズをホストバスアダプタ（ＨＢＡ）ドライバによって課されたコマンドキュー制限に従って制限する。

【００４０】

ビジー環境においてホストが示す通常の挙動は、ホストがバースト要求（ホストのキューで制限されたもの）を送信した後、それらの要求がアレイによって処理されるのを待つというものになる。ディスクアレイがそれらの要求を処理するのに従って、ホスト側のコマンドキューが減少し、ホストはより多くの要求を送信できるようになる。ホスト側のコマンドキューサイズを低減することを含む策により、ＱＦＵＬＬの戻りステータスで拒絶されるコマンド数を最低限に抑えることができる。

【００４１】

実際には、ホストの要求速度が一定でないように、アレイのＩ／Ｏ処理速度も一定ではない。上記のように、本発明の原理に基づくと、記憶装置は、システムの必要性に合わせてサービスキュー制限値を調節することが可能になる。さらに、ホスト装置（単数の場合も複数の場合もある）のコマンドキューも、システムの必要性に合わせて調節することが可能である。

【００４２】

図３は、本発明の他の実施形態によるシステムを示す。このシステムは、少なくとも１つのホスト装置と少なくとも１つの記憶装置とを有し、ホスト装置（単数の場合も複数の場合もある）のコマンドキューのサイズを調節して、システムを均等にすることができる。

【００４３】

たとえば、キュー（１０３）を有する特定のディスクアレイ（１０１）が、７５０コマンドの論理サイズ（１０４）を有するものと仮定する。この容量にも関らず、この特定のディスクアレイのキュー深さ、すなわち接続されたホスト装置（１１０）によりキュー（１０３）に配置される要求は、通常の動作中に約１００コマンドであり、適度な数のコマンドを完了するのに０．２秒よりも長くかかることが分かっている。

【００４４】

この実施例では、論理キューサイズ（１０４）をデフォルトの７５０コマンドから実際の使用数である１００コマンドに変更することにより、ディスクアレイ（１０１）のコマンドレイテンシ曲線がさらに広がるのを防止することができる。そして、ホスト（１１０）がもっとアクティブになると、アレイコントローラ（１０２）は、この活力の増加によって生じるコマンドを、ＱＦＵＬＬステータスの指示を用いて拒絶することにより応答する。

【００４５】

本発明の原理に基づくと、ホストドライバ（３０１）は、ホスト側のコマンドキュー（３０２）を減少させることにより応答することが好ましい。これにより、ホスト（１１０）から記憶装置（１０１～１０３）に送信されるコマンドの速度が減少する。

【００４６】

平衡状態に達した後、アレイ（１０１）は、作業負荷が増大する前と同じコマンドレイテンシレベルでコマンドを処理することになる。ホスト側では、システムコールインタフェースのブロッキングの性質により、アプリケーションに要求速度を「キュー満杯閾値」で管理されたレベルに強制的に維持させる。このシナリオは、応答時間を所望のレベルに維持するが、要求間の最小時間は増大する。この要求間の時間の増大は、長い目で見ればほとんどのユーザが満足しないことになるということの兆候である。もっと良い解決方法は、構成にさらなる記憶処理能力を追加することである。

【００４７】

また、この「キュー満杯閾値」の調節を利用して、長いドライバ関連タイムアウトが発生する割合に影響を与えることもできる。特定の作業負荷について、オペレーティングシステム（Ｏ／Ｓ）ドライバ（３０１）が５秒毎に１回の割合でタイムアウトイベントを報告している間、記憶装置（１０１～１０３）が４００コマンドの論理キュー深さを維持しているものと仮定する。その場合、コマンドレイテンシ分布の末尾を短くするためには、キューの論理サイズ（１０４）を３００まで縮小することが好ましい。

【００４８】

すると、アレイコントローラ（１０２）は、ＱＦＵＬＬステータスによって大量のコマンドを拒絶することになる。ホスト側のドライバ（３０１）は、ＱＦＵＬＬ戻りステータスがなくなるまでホスト側のキュー（３０２）の深さを縮小することにより、ＱＦＵＬＬステータスに応じることが好ましい。今度は記憶装置（１０１～１０３）がもっと短いコマンドキュー（１０４）で動作しているので、著しい待ち時間（例えば３０秒など）をもつコマンドの割合が大幅に減少するはずである。

【００４９】

ホスト側のドライバ（３０１）は、ホスト側のコマンドキュー（３０２）の深さを再度増大させることにより、記憶装置（１０１～１０３）を時々検査する。
これを行なう場合、ホスト側のドライバ（３０１）がホスト側のキュー（３０２）の深さを再度縮小させることを判断するまで、ＱＦＵＬＬ応答が生じることになる。

【００５０】

先の説明は、本発明の例示及び説明のために提供したものに過ぎない。その説明は、網羅的なものにすることも、本発明を開示したいずれかの形態そのままに制限することも意図していない。上記の示唆を考慮して、多数の変更および変形が可能である。

【００５１】

好ましい実施形態は、本発明の原理とその実際の使用を最もよく例示する目的で選択され、説明されている。先の説明は、他の当業者が本発明を様々な実施形態で意図する特定の用途に適するように様々な変更を加えて最もよく利用できるようにすることを意図している。本発明の範囲は、特許請求の範囲で規定されることを意図している。

【図面の簡単な説明】

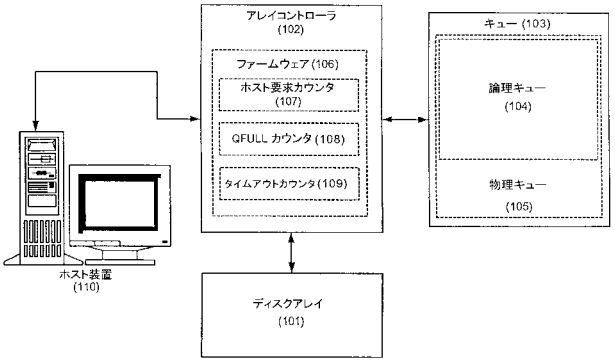
【００５２】

【図１】本発明の一実施形態による改良されたコンピュータシステムを示すブロック図であり、該システムは、少なくとも１つのホスト装置と、調節可能なキューを有する少なくとも１つのデータ記憶装置とを有する。

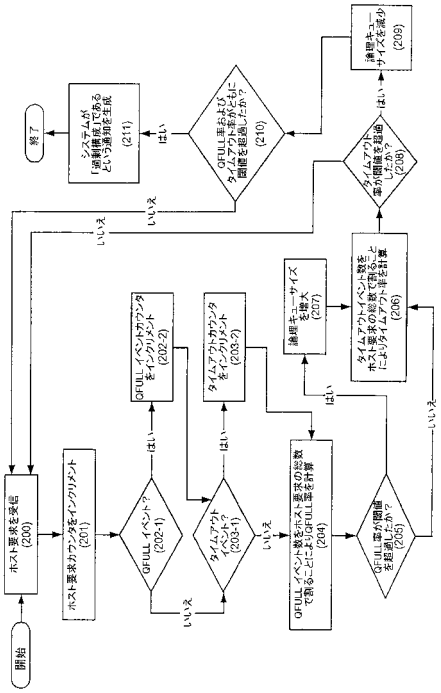
【図２】本発明の一実施形態による、ホスト装置とデータ記憶装置とを備えたコンピュータシステムを動作させる方法を示すフロー図である。

【図３】本発明の他の実施形態によるコンピュータシステムを示す図であり、該システムは、少なくとも１つのホスト装置と少なくとも１つの記憶装置とを有し、ホスト装置のコマンドキューのサイズを調節して、システムを均等にすることができるようになっている。

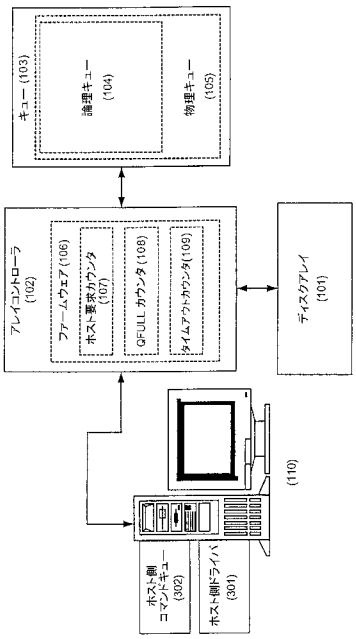
【図１】


アレイコントローラ
(102)
ファームウェア(106)
ホスト要求カウンタ
(107)
QFULL カウンタ(108)
タイムアウトカウンタ(109)
キュー (103)
論理キュー
(104)
物理キュー
(105)
ホスト装置
(110)
ディスクアレイ
(101)

【図２】


開始
ホスト要求を受信
(200)
ホスト要求カウンタをインクリメント
(201)
QFULL イベント?
(202-1)
QFULL イベントカウンタをインクリメント
(202-2)
タイムアウトイベント?
(203-1)
タイムアウトカウンタをインクリメント
(203-2)
QFULL イベント数をホスト要求の総数で割ることにより QFULL率を計算
(204)
QFULL率が閾値を超えたか?
(205)
タイムアウトイベント数をホスト要求の総数で割ることにより タイムアウト率を計算
(206)
論理キューサイズを減少
(209)
タイムアウト率が閾値を超えたか?
(208)
論理キューサイズを増大
(207)
QFULL 率またはタイムアウト率が閾値を超過したか?
(210)
システムが「過剰構成」であるという通知を生成
(211)
終了
はい
いいえ

【図３】


キュー (103)
論理キュー
(104)
物理キュー
(105)
アレイコントローラ
(102)
ファームウェア (106)
ホスト要求カウンタ
(107)
QFULL カウンタ (108)
タイムアウトカウンタ(109)
ディスクアレイ
(101)
(110)
ホスト側
コマンドキュー
(302)
ホスト側ドライバ
(301)

フロントページの続き

(72)発明者　ランディ・ジェイ・マシューズ
アメリカ合衆国アイダホ州８３７１３，ボイス，ヒッコリー・ナット・ストリート・１１２４５
(72)発明者　マーク・イー・レフェブル
アメリカ合衆国アイダホ州８３７１３，ボイス，ウエスト・アナブルック・ドライブ・１３４０１
(72)発明者　リチェル・エル・アールバーズ
アメリカ合衆国アイダホ州８３６１６，イーグル，ウエスト・キャンポ・レーン・１９５１
(72)発明者　ウェイド・エイ・ドルフィン
アメリカ合衆国アイダホ州８３６１９，イーグル，ウエスト・キャンポ・１９５１
(72)発明者　ダグラス・エル・ボイト
アメリカ合衆国アイダホ州８３７０２，ボイス，ノース・トゥエンティフォース・３０３０

Ｆターム(参考) 5B065 BA01　CA15　CA30　CC08　CH18　EK06