

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4265728号
(P4265728)

(45) 発行日 平成21年5月20日 (2009.5.20)

(24) 登録日 平成21年2月27日 (2009.2.27)

(51) Int.Cl.

F I

G 0 6 F 12/00 (2006.01)

G 0 6 F 12/00 5 3 3 F

請求項の数 18 外国語出願 (全 30 頁)

(21) 出願番号	特願2001-197684 (P2001-197684)	(73) 特許権者	000005108
(22) 出願日	平成13年6月29日 (2001.6.29)		株式会社日立製作所
(65) 公開番号	特開2002-149463 (P2002-149463A)		東京都千代田区丸の内一丁目6番6号
(43) 公開日	平成14年5月24日 (2002.5.24)	(74) 代理人	110000279
審査請求日	平成17年3月28日 (2005.3.28)		特許業務法人ウィルフォート国際特許事務所
(31) 優先権主張番号	09/606403	(72) 発明者	児玉 昇司
(32) 優先日	平成12年6月30日 (2000.6.30)		アメリカ国カリフォルニア州サンノゼ
(33) 優先権主張国	米国 (US)		レーン323 エランビレッジ335
		(72) 発明者	山本 彰
			アメリカ国カリフォルニア州クパチーノ
			ルッペルプレイス1202
		(72) 発明者	山神 憲司
			アメリカ国カリフォルニア州ロスガトス
			カイルニベル108

最終頁に続く

(54) 【発明の名称】 データ・サーバ・システム内のデータの連続更新方法

(57) 【特許請求の範囲】

【請求項1】

コンテンツ・サーバ・システム内のデータを更新する方法であって、

前記コンテンツ・サーバ・システムは、第1、第2、および第3のデータ・ストアをそれぞれ有するストレージ・デバイスと、前記ストレージ・デバイスとデータ通信を行うデータ・スイッチと、前記データ・スイッチと通信し、前記第1のデータ・ストアからデータを読み出すことのできるデータ・サーバと、前記第2のデータ・ストアとデータの交換を行うために、前記データ・スイッチとデータ通信を行うオーサリング・サーバとを備えており、

前記ストレージ・デバイスにより実行されるステップであって、前記第1、第2、および第3のデータ・ストアのそれぞれが同一のデータを記憶するように、前記システムが起動される場合に同期させるステップと、

前記データ・サーバに前記第1のデータ・ストアをマウントさせるステップと、

前記データ・サーバが、クライアントの読み出し要求に回答して、前記第1のデータ・ストアからデータを読み出すステップと、

前記オーサリング・サーバが、前記第2のデータ・ストア内のデータを更新する第1の更新ステップと、

前記オーサリング・サーバからの同期信号を前記ストレージ・デバイスが受け取るステップと、

前記同期信号の受け取りに応じて、前記ストレージ・デバイスが、前記第2のデータ・

10

20

ストアのデータと前記第 3 のデータ・ストアのデータとを一致させるべく、前記第 2 のデータ・ストアから前記第 3 のデータ・ストアにデータのコピーを行うステップと、

前記第 2 のデータ・ストアからのデータがコピーされた前記第 3 のデータ・ストアを、前記データ・サーバにマウントさせ、かつ、前記第 1 のデータ・ストアを前記データ・サーバからアンマウントさせるステップと、

その後を受け取ったクライアントからの読み出し要求に応じて、前記データ・サーバが、前記第 3 のデータ・ストアからデータを読み出すステップと、
を有することを特徴とするデータ更新方法。

【請求項 2】

請求項 1 記載のデータ更新方法であって、

前記コピーを行うステップは、前記データ・サーバが前記第 3 のデータ・ストアからデータを読み出すステップとともに行うことを特徴とする方法。

【請求項 3】

請求項 1 記載のデータ更新方法であって、

前記データ・サーバに前記第 3 のデータ・ストアからデータを読み出させるステップでは、前記ストレージ・デバイスは、前記第 1 の更新ステップにより更新された前記第 2 データ・ストア内の更新データが、前記コピーを行うステップにより前記第 2 のデータ・ストアから前記第 3 のデータ・ストアにコピーされているか否かを判定し、前記更新データが前記第 3 のデータ・ストアにコピーされている場合には、前記データ・サーバに前記第 3 のデータストアから前記更新データを読み出させ、前記更新データが前記第 3 のデータ・ストアにコピーされていない場合には、前記データ・サーバに前記第 2 のデータ・ストアから前記更新データを読み出させる、ことを特徴とするデータ更新方法。

【請求項 4】

請求項 1 記載のデータ更新方法であって、

前記第 3 のデータ・ストアからデータを読み出すステップは、前記クライアントからの読み出し要求を満たすために、更新されたデータを前記第 2 のデータ・ストアから読み出すステップ、および前記更新されたデータを前記第 3 のデータ・ストアに書き込むステップを含むことを特徴とするデータ更新方法。

【請求項 5】

請求項 1 記載のデータ更新方法であって、

前記第 1 のデータ・ストアからデータを読み出すステップは、データ・スイッチを介して前記データを前記データ・サーバに伝送するステップを含むことを特徴とするデータ更新方法。

【請求項 6】

請求項 5 記載のデータ更新方法であって、

前記コピーを行うステップは、前記データ・スイッチを使用せずに、前記ストレージ・デバイス内で、前記第 2 のデータ・ストアから前記第 3 のデータ・ストアにデータを直接転送することにより実行される、ことを特徴とするデータ更新方法。

【請求項 7】

請求項 5 記載のデータ更新方法であって、

前記データ・サーバが前記第 3 のデータ・ストアからデータを読み出すステップは、前記データ・スイッチを介して前記データを伝送するステップを含むことを特徴とするデータ更新方法。

【請求項 8】

請求項 7 記載のデータ更新方法であって、

前記第 1 の更新ステップは、前記データ・スイッチを介してデータを伝送するステップを含むことを特徴とするデータ更新方法。

【請求項 9】

請求項 1 記載のデータ更新方法であって、

前記第 1 のデータ・ストア内の前記データの一部は、キャッシュ・メモリ内に含まれ、

10

20

30

40

50

前記データを読み出すステップは、前記キャッシュ・メモリからデータを読み出すステップを含むことを特徴とするデータ更新方法。

【請求項 10】

請求項 9 記載のデータ更新方法であって、

さらに、前記第 1 のデータ・ストアがアンマウントされた場合に、前記データ・サーバが、前記キャッシュ・メモリをフラッシュするステップを含むことを特徴とするデータ更新方法。

【請求項 11】

ストレージ・デバイスを有するデータ・サーバにおける、前記ストレージ・デバイスに含まれているデータを、クライアントに対するサービスを中断することなくデータを更新する方法において、

前記ストレージ・デバイスにより実行されるステップであって、前記ストレージ・デバイス内の第 1、第 2、および第 3 のデータ・ストアのそれぞれが同一のデータを記憶するように、前記システムが起動される場合に同期させるステップと、

第 1 のデータ・プロセッサが、データ・スイッチを介して前記第 1 のデータ・ストアから前記データを読み出すステップと、

第 2 のデータ・プロセッサが、前記第 2 のデータ・ストアに前記データ・スイッチを介してデータを送信することにより、前記第 2 のデータ・ストア内のデータを更新させるステップと、

前記第 2 のデータ・プロセッサにより第 1 の信号を生成させるステップと、

前記第 1 の信号に応答して、前記ストレージ・デバイスが、前記第 2 のデータ・ストアから第 3 のデータ・ストアにデータを書き込むステップと、

前記第 2 のデータ・プロセッサにより第 2 の信号を生成するステップと、

前記第 2 の信号に応答して、前記第 1 のデータ・プロセッサが、前記データ・スイッチを介して該データを、前記第 2 のデータ・ストアから、もしくは前記第 3 のデータ・ストアから獲得するステップと、

を包含することを特徴とするデータ更新方法。

【請求項 12】

クライアントに対してデータを提供するためのサーバ・システムであって、

第 1、第 2、および第 3 のデータ・ストアを有するストレージ・デバイスと、

前記ストレージ・デバイスとデータ通信を行うデータ・スイッチと、

前記データ・スイッチと通信し、前記複数のデータ・ストアのうちの第 1 のデータ・ストアからデータを読み出すことのできる第 1 のデータ・サーバと、

前記複数のデータ・ストアのうちの第 2 のデータ・ストアとデータの交換を行うために、前記データ・スイッチとデータ通信を行うオーサリング・サーバとを備え、

前記ストレージ・デバイスは、前記第 1、第 2、および第 3 のデータ・ストアのそれぞれが同一のデータを記憶するように、前記システムが起動される場合に同期させ、

前記第 1 のデータサーバに前記第 1 のデータ・ストアをマウントさせ、

クライアントの読出し要求に応じて、前記第 1 のデータ・サーバにより前記データ・スイッチを介して前記第 1 のデータ・ストアからデータを読み出させ、

前記オーサリング・サーバにより前記データ・スイッチを介して前記第 2 のデータ・ストア内のデータを更新させ、

前記ストレージ・デバイスは、前記オーサリング・サーバから第 1 の信号を受け取った場合に、前記第 2 のデータ・ストアのデータのうち前記オーサリング・サーバにより更新されたデータを前記第 3 のデータ・ストアにダイレクトに転送して記憶させ、

前記第 1 のデータ・サーバは、前記オーサリング・サーバから第 2 の信号を受け取った場合に、前記第 3 のデータ・ストアをマウントして、前記第 3 のデータ・ストアからデータを読み出す、サーバ・システム。

【請求項 13】

データ・サーバ・システム内のデータを更新するデータ更新方法において、

前記データ・サーバ・システムは、第1および第2のデータ・ストアをそれぞれ有するストレージ・デバイスと、前記ストレージ・デバイスとデータ通信を行うデータ・スイッチと、前記データ・スイッチと通信し、前記第1のデータ・ストアを更新させるオーサリング・サーバと、前記データスイッチと通信し、前記第2のデータ・ストアからデータを読み出すことのできるデータ・サーバと、とを備えており、

クライアントからのデータの読み出し要求に応えるために前記データ・サーバが前記第2のデータ・ストアのデータを読み出し、前記オーサリング・サーバが前記第1のデータ・ストアを更新させるステップと、

前記オーサリング・サーバが第1の信号を生成するステップと、

クライアントからのデータの読み出し要求に応えるために、前記ストレージ・デバイスが、前記第1の信号にตอบสนองして、前記第1のデータ・ストアのデータを読み出し、かつ、そのデータを前記第1のデータ・ストアから前記第2のデータ・ストアにコピーするステップと、

前記第1の信号にตอบสนองして、前記データ・サーバが、前記第2のデータ・ストアを更新するためにそれに対する読み出しおよび書き込みを行うステップと、
を包含することを特徴とするデータ更新方法。

【請求項14】

請求項13記載データ更新方法であって、前記データ・サーバが前記第2のデータ・ストアに対する読み出しおよび書き込みを行うステップは、前記第2のデータ・ストアに記憶されているデータについての未処理の要求が全て処理されるまで待機することを特徴とするデータ更新方法。

【請求項15】

請求項13記載データ更新方法であって、前記データ・サーバが前記第2のデータ・ストアに対する読み出しおよび書き込みを行うステップでは、前記ストレージ・デバイスは、前記オーサリング・サーバにより更新された前記第1のデータ・ストア内の更新データが前記第2のデータ・ストアにコピーされているか否かを判定し、前記更新データが前記第2のデータ・ストアにコピーされている場合は、前記データ・サーバに前記第2のデータ・ストアから前記更新データを読み出させ、前記更新データが前記第2のデータ・ストアにコピーされていない場合は、前記第1のデータ・ストアから前記更新データを読み出して前記第2のデータ・ストアにコピーさせる、ステップを含むことを特徴とするデータ更新方法。

【請求項16】

複数のクライアントに対してデータを配信するためのシステムにおいて、第1および第2のデータ・ストアを有するストレージ・サブシステムと、データ・スイッチ・サブシステムと、前記データ・スイッチを介して前記第1のデータ・ストアと通信し、前記第1のデータ・ストアからデータを読み出すことによって、クライアントからの読み出し要求を満たすべく構成された第1のプロセッシング・サブシステムと、

前記データ・スイッチを介して前記第2のデータ・ストアとデータ通信し、前記第2のデータ・ストアに記憶される更新済みデータを生成するべく構成された第2のプロセッシング・サブシステムと、を備え、

前記第2のプロセッシング・サブシステムは、前記第2のデータ・ストアに記憶されている前記更新済みデータが前記第1のデータ・ストアにコピーされているか否かを判定し、前記更新済みデータが前記第1のデータ・ストアにコピーされていない場合には、前記クライアントからの前記更新済みデータの読み出し要求を、前記第2のデータ・ストアの前記更新済みデータを読み出すことによって処理しなければならないことを示す第1の信号と、前記第1のデータ・ストアからデータを読み出す未処理の読み出し要求がないことを示す第2の信号と、を生成する手段を有し、

前記第1のプロセッシング・サブシステムは、さらに前記第1の信号の検出にตอบสนองして、前記第1のデータ・ストアとのデータ通信を中断し、かつ前記第2のデータ・ストアと

10

20

30

40

50

のデータ通信を設定する手段を有し、

前記ストレージ・サブシステムは、前記第2の信号の検出に応じて、前記データ・スイッチを使用せずに、前記ストレージ・サブシステム内で、前記第2のデータ・ストア内の前記更新済データを前記第1のデータ・ストアに転送する手段を有する、
ことを特徴とするシステム。

【請求項17】

請求項16記載のシステムであって、

前記第2のプロセッシング・サブシステムは、さらに前記第2の信号が生成された場合に、前記第1のデータ・ストア内のデータを更新する手段を有し、

前記ストレージ・サブシステムは、前記第2のデータ・ストアに記憶されている前記更新済データが前記第1のデータ・ストアにコピーされているか否かを判定し、前記更新済データが前記第1のデータ・ストアにコピーされていない場合には、前記第2のデータ・ストアから前記更新済データを読み出すことにより、前記第2のプロセッシング・サブシステムからの前記第1のデータ・ストアに対する読み出し要求を処理する手段を有することを特徴とするシステム。

【請求項18】

請求項16記載のシステムであって、前記データ・スイッチは、ファイバチャネルスイッチであって、前記第1のプロセッシング・サブシステムは、さらに、クライアントからの、前記第1のデータ・ストアについての未処理の読み出し要求が全て処理されるまで、前記第1のデータ・ストアとのデータ通信の中断を遅延する手段を有することを特徴とするシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、概してクライアント・サーバ・システムに関し、より詳細に述べれば、ネットワーク内におけるクラスタ化されたデータ・コンテンツ・サーバに関する。

【0002】

【従来の技術】

クラスタ化されたデータ・サーバ環境においては、各サーバが、ローカル・ディスク・ドライブ内にそのコンテンツをストアしている。たとえば、データ・サーバとしてウェブ・サーバ、ドキュメント・サーバ、イメージ・サーバ、ビデオ・サーバ等々が考えられる。各ローカル・ディスク・ドライブは、オリジナル（またはマスタ）ディスク・ドライブのミラーである。マスタ・ディスク内のウェブ・コンテンツは、オーサリング・サーバによって作成され、更新される。新しいコンテンツまたは更新後のコンテンツは、オーサリング・サーバから各ウェブ・サーバにネットワークを介して転送される。この種の転送の数が多すぎるとき、および/または転送されるデータのサイズが大きすぎるとき、データの転送に長時間を要することになる。これは、受信側ウェブ・サーバのCPU負荷を増加させ、その結果、そのコンテンツの更新が行なわれている間は、到来したHTTP要求をウェブ・サーバが処理できなくなる。この理由により、通常は、オーサリング・サーバからの更新を日中に行うことができない。インターネット・トラフィックは地球規模であり、世界中のどこかに必ず「日中」に該当するところがあることから、問題はさらに悪化する。

【0003】

一方、共有ファイル・システムを実装しているときには、別の問題が発生する。共有ファイル・システムにおいては、すべてのウェブ・サーバが1つのディスク・ドライブ（より一般的には、ディスク・ドライブのシステム）を共有し、ファイル・サーバがファイル・システムのメタ・データをマネージする。しかしながら、この共有ファイル・システムのファイル・サーバが、信頼性ならびにパフォーマンスの両方の意味においてボトルネックとなる。ファイル・サーバなしでは、ウェブ・サーバが共有ディスク・ドライブからコンテンツを読み出すことが不可能であり、すべてのウェブ・サーバは、ファイルのオープン

を試みるごとに、ファイル・サーバと通信しなければならない。結局、これが大きな待ち時間を招く結果となり、どのような長さであっても通常はエンド・ユーザによって許容される類いのものではない。

【 0 0 0 4 】

【 発明が解決しようとする課題 】

データ・サーバのコンテンツのリアルタイム更新を可能にするソリューションが必要とされている。特に、データ・サーバがクライアントの要求を処理しているとき、この能力を提供し得ると望ましい。また、システムのパフォーマンスを低下させることなく、リアルタイムのコンテンツ更新が提供されることが望まれている。

【 0 0 0 5 】

【 課題を解決するための手段 】

クライアントに対するサービスに影響を与えることなくデータ・サーバにおけるデータを連続更新するための方法およびシステムは、第 1 のデータ・ストアからの読み出しによって、データを求めるクライアントの要求を満たすステップを含む。同時に、システムのブート・アップ時には第 1 のデータ・ストアと同一であった第 2 のデータ・ストアを更新する。随時、第 2 のデータ・ストア内の新しいコンテンツを第 1 のデータ・ストアにコピーすることが望ましいときには、第 2 のデータ・ストアからのデータが第 3 のデータ・ストアにコピーされる。またその後の、データを求めるクライアントの要求は、第 3 のデータ・ストアからの読み出しによって満たされる。さらに、第 2 のデータ・ストア上において、作成 / 更新アクティビティの別のラウンドが開始される。

【 0 0 0 6 】

本発明の 1 つの側面は、データ・スイッチを介したデータの転送を含むが、コピー・オペレーションの実行においては、このデータ・スイッチが除かれる。本発明の別の側面は、データを求めるクライアントの要求を満たすための第 3 のデータ・ストアからの読み取りオペレーションを、実際には第 2 のデータ・ストアからの読み取りオペレーションとする必要があるか否かの決定を含む。本発明のさらに別の側面においては、作成 / 更新の次のラウンドの間における第 2 のデータ・ストアに対する書き込みオペレーションに、第 2 のデータ・ストアから第 3 のデータ・ストアへの書き込みオペレーションを先行させなければならぬか否かの決定を含む。

【 0 0 0 7 】

本発明の別の実施態様においては、データを求めるクライアントの要求が第 1 のデータ・ストアからの読み出しによって満たされる。同時に、第 2 のデータ・ストア上において作成 / 更新アクティビティが生じる。更新が望まれる場合には、その後のデータを求めるクライアントの要求が第 2 のデータ・ストアからの読み出しによって満たされる。次の作成 / 更新アクティビティのラウンドは、第 1 のデータ・ストアに対する読み出しおよび書き込みによって実行される。本発明のこの実施態様の 1 つの側面は、第 1 のデータ・ストアの読み出しおよび書き込みに先行して、クライアントが開始した未決定の第 1 のデータ・ストアからの読み出しオペレーションがすべて完了するまでの待機が存在することである。その種のクライアントが開始した未決定の読み出しオペレーションがすべて完了すると、コピー・オペレーションが開始されて、第 2 のデータ・ストアのコンテンツが第 1 のデータ・ストアにコピーされる。本発明の別の側面は、データ・スイッチを介してデータの転送が行なわれるが、コピー・オペレーションについては、データ・スイッチを除いたところに生じることである。

【 0 0 0 8 】

本発明の教示は、以下の図面を参照した詳細な説明を考察することによって容易に理解を得ることができよう。

【 0 0 0 9 】

【 発明の実施の形態 】

図 1 は、本発明に従ったデータ・サーバ・システムの実施態様を示している。データ・サーバの 1 つの応用は、図に示すようなウェブ・サーバ・マシン内にある。ただしこのウェ

10

20

30

40

50

ブ・サーバ・マシンは、本件出願の時点において本発明を実施する最良の形態を説明するためのピークルとしての作用を提供する単なる例に過ぎないことに注意する必要がある。また、本発明は任意のデータ・サーバ・システムに容易に適合され得ることを理解する必要がある、たとえばデータ・サーバは、ウェブ・サーバ・マシン、ドキュメント・サーバ、イメージ・サーバ・ビデオ・サーバ等々とすることができる。

【0010】

図1に示されるように、ワールド・ワイド・ウェブ(WWW、またはウェブ)サーバ環境は、1ないしは複数のクライアント010101、010102、...、010103を含み、そのそれぞれがウェブ・ブラウジング・ソフトウェアを有している。これらのクライアントは、1ないしは複数のウェブ・サーバ・マシンに要求を送信し、コンテンツ、たとえばHTMLデータ、イメージ・データ、オーディオ・データ等を獲得する。続いて、受信したコンテンツをウェブ・ブラウジング・ソフトウェアがそれらのブラウザ上に表示する。

10

【0011】

各クライアントは、インターネット010201によりウェブにアクセスすることができる。これは通常、公衆交換網、つまり電話システムによってアクセスされる。それに代えて、ケーブル・テレビジョン・オペレータが、ケーブル・モデム・サービスを介した高速アクセスを提供することもあり、それにおいては、テレビジョン・ケーブル・ネットワークの高いデータ容量を使用してインターネット・トラフィックが搬送される。クライアント010101、010102、...、010103は、電話接続、ケーブル接続、ワイヤレス・リンク等を介したインターネットに対する接続011201、011202、...、011203を有している。

20

【0012】

ウェブ・サーバ・マシン・サイトにおいては、負荷バランサ010301がクライアントからの要求を受信する。「ウェブ・サーバ」は、クライアントに対して許容可能な応答時間を提供するために、通常、複数のウェブ・サーバ・マシンを包含している。負荷バランサは、複数のウェブ・サーバ・マシンの1つを選択して受信した要求を渡し、選択したウェブ・サーバ・マシンに処理を行わせる。このようにすることによって、負荷バランサは、ウェブ・サーバ・マシン間のCPU負荷がより均一となるように分散させている。

30

【0013】

ウェブ・サーバ・マシン・サイトには、さらにオーサリング・サーバ010401が備わっている。このコンポーネントは、ウェブ・サイトが提供するコンテンツの作成および更新を行う。前述したように、通常ウェブ・サイトは、複数のウェブ・サーバ・マシン010501、...、01050Nを包含している。一般にこれらのマシンは、UNIX(ユニックス)ベースのシステムまたはUNIX(ユニックス)から派生したシステムを使用しており、その理由としては、これらのシステムが時の試練を経ていること、堅牢であること、サーバ・システムに求められるスループットを提供し得ることが挙げられる。別のシステム、たとえばWindows NT(ウインドウズNT)も考慮に入れられる。

【0014】

各ウェブ・サーバ・マシンは、通常、2つのコンポーネントを包含している。第1のコンポーネントはHTTP(ハイパーテキスト・トランスファ・プロトコル)デーモン011501、...、01150Nである。HTTPデーモンは、ウェブ・サーバ・マシン上で走るプロセスであり、ウェブ・サイトのコンテンツに関するユーザの要求を処理する。まずHTTPデーモンは、ブラウザ中のクライアントからの要求を受け取る。続いてHTTPデーモンは、クライアントが要求を行っているコンテンツが何かを解釈する。次にHTTPデーモンは、データ・ストアにアクセスし、そこからコンテンツを読み出す。最後にHTTPデーモンは、アクセスしたコンテンツを、インターネットを介して要求を発しているクライアントに送信し、それによりデータに対するクライアントの要求を満たす。

40

【0015】

各ウェブ・サーバ・マシンの第2のコンポーネントは、リマウント・デーモン01160

50

1、．．．、01160Nである。各リモウント・デーモンは、オーサリング・サーバと通信する。このコンポーネントの役割については後述する。

【0016】

各ウェブ・サーバ・マシン010501、．．．、01050Nは、ローカル・ファイル・システムを介してデータ・ストア011101、011102をマウントすることができる。つまり、各ウェブ・サーバ・マシンは、たとえば、ファイル・サーバを使用する共有ファイル・システムに見られるような通信のオーバーヘッドをまったく伴うことなくダイレクトにデータ・ストアからのデータにアクセスすることができる。

【0017】

データ・ストレージ・システム010801に対するウェブ・サーバ・マシン010501、．．．、01050Nおよびオーサリング・サーバ010401の接続には、高速データ・スイッチ010601が使用される。好ましい実施態様においては、データ・スイッチが米国規格協会（ANSI）によって定義されているFibre Channel（ファイバ・チャンネル）標準に準拠する。ただし、本発明がFibre Channel（ファイバ・チャンネル）スイッチの使用によって限定されないことを理解する必要がある。そのほかのデータ・スイッチング・テクノロジーも使用することができる；たとえば、ギガビット・イーサネット・データ・スイッチおよびATMスイッチングをその可能性として挙げられる。しかしながら、Fibre Channel（ファイバ・チャンネル）テクノロジーは、その標準が高速かつ堅牢なデータ・スイッチング・アプリケーションを念頭に定義されていることから好ましい。

【0018】

ストレージ・デバイス010801は、マスタ・ディスク011001および2つのローカル・ディスク011101、011102を含む複数のディスク・ドライブを包含している。各ウェブ・サーバ・マシン010501、．．．、01050Nは、そのファイル・システム上に、いずれのローカル・ディスクもマウントすることが可能であり、それを行うことによって、各サーバは、同一のディスクを共有することができる。伝統的なWWW環境においては、各ウェブ・サーバ・マシンが独自のローカル・ディスクを有しており、オーサリング・サーバもまた、独自のローカル・ディスクを有している。したがって、ローカル・ディスク内のウェブ・サーバ・マシンのコンテンツを更新するときには、オーサリング・サーバが、ネットワーク、たとえばイーサネットを介し、ファイバ経由でウェブ・サーバ・マシンに対し、新しい、あるいは更新済みのコンテンツを送信しなければならない。このアプローチは、各ウェブ・サーバ・マシンからのCPUのオーバーヘッドを招き、結果的にそれがシステムのパフォーマンスを低下させることになる。この理由から、更新を随時行うことができず、1日のうちの特定の時間帯に限りそれが行なわれている。

【0019】

ストレージ・デバイス010801は、さらにディスク・ドライブのマネージを行うディスク・コントローラ010901を含んでいる。ディスク・コントローラは、サーバからのI/O要求を処理する。各サーバは、データ・スイッチ010601を介してディスク・コントローラに結合されており、その結果、各サーバは、見かけ上、ディスク・コントローラに対する単独のアクセスを有する。

【0020】

ストレージ・デバイスは、マスタ・ディスク011001を含んでいる。オーサリング・サーバは、新しいコンテンツを作成し、マスタ・ディスク上に収められている現存するコンテンツを更新する。マスタ・ディスクのコンテンツは、適宜、たとえばスケジュールに従って、あるいはシステム管理者による決定に応じてローカル・ディスクにコピーされる。

【0021】

ローカル・ディスク[1]およびローカル・ディスク[2]として識別されているローカル・ディスク011101、011102は、各ウェブ・サーバ・マシンのファイル・シ

10

20

30

40

50

ステム上にマウントされる。本発明によれば、2つのローカル・ディスクのうち的一方だけが、常時すべてのウェブ・サーバ・マシンにマウントされる。

【0022】

このストレージ・デバイスにはさらに、再同期デーモン010902が含まれている。このコンポーネントは、ディスクのスナップショットを作成する。本発明によれば、再同期デーモンが更新済みのデータをマスタ・ディスクからローカル・ディスクの1つにコピーする。

【0023】

さらに図1を参照すると、上記のコンポーネント間の相互接続が示されている。上記のコンポーネントの間には、いくつかのネットワークが存在する。たとえば、ブラウザ中のクライアント010101、...、010103は、通信回線011201、...、011203を介してインターネットに接続されている。負荷バランサ010301は、光ファイバ回線、ケーブル・モデム接続、DSL回線等の高速リンク011204を介してインターネットに接続されている。また負荷バランサは、イーサネット接続011205を介してウェブ・サーバ・マシンおよびオーサリング・サーバに接続されている。各サーバは、Fibre Channel（ファイバ・チャンネル）011206、011301、...、01130N、011207により、データ・スイッチ010601を介してストレージ・デバイス010801に接続されている。ストレージ・デバイス内においては、ディスク・コントローラ010901が、SCSI 011208、011209、011210を介してディスク・ドライブに接続されている。

【0024】

ここで簡単に図10～13を参照するが、上記のコンポーネントによって各種の内部データ変数が維持されることがわかる。オーサリング・サーバ010401は、アンマウント・ディスク番号DN 160101（図13）を管理している。このデータ変数は、ローカル・ディスク[DN]が、いずれのウェブ・サーバ・マシンによってもマウントされていないことを示す。

【0025】

各ウェブ・サーバ・マシン010501、...、01050Nは、そのルート・ディレクトリのディスク番号、RootDN 110101（図11）を管理している。この変数は、各ウェブ・サーバ・マシンが、ローカル・ディスク[RootDN]からコンテンツの読み出しを行っていることを示す。また各ウェブ・サーバ・マシンは、各ローカル・ディスクに関する完了していない要求の数を管理する。たとえば、ウェブ・サーバ・マシン010501によって管理されるReqN[1] 100101（図10）は、ローカル・ディスク[1]に関する完了していない要求の数を表し、それにおいて要求は、ウェブ・サーバから発行される。ウェブ・サーバ・マシンは、そのReqN[i]がゼロである場合に限ってローカル・ディスク[i]をアンマウントすることができる。

【0026】

ストレージ・デバイス010801は、ビットマップのセットBM1[1][x]、BM2[1][x]、BM1[2][x]、およびBM2[2][x]（120201、130201、140201、150201；図12）を管理している。これらのビットマップは、2つのディスク・ドライブの間、すなわちマスタ・ディスクおよびローカル・ディスク[1]またはローカル・ディスク[2]のいずれかの間におけるミラーリングに使用される。MaxDBNは、ローカル・ディスクの有するディスク・ブロックの最大数である。本発明においては、すべてのディスク・ドライブの有するディスク・ブロックが同数となる。

【0027】

BM1ビットマップは、マスタ・ディスクが、ローカル・ディスクにまだコピーされていない新しい情報を有するか否かを示し、ここでは情報ありを、慣習に従ってビット値「1」により示す。つまり、BM1[DN][x]が「1」であれば、マスタ・ディスクのブロックxが新しい情報を含んでおり（あるいは新しい情報が書き込まれている）、かつロ

10

20

30

40

50

ーカル・ディスク [D N] のそのブロック x を、最終的に、すなわち、オーサリング・サーバが更新オペレーションの実行を要求したとき、更新して新しい情報を持たせる必要があることを示す（本発明のこの実施態様においては、D N が「 1 」または「 2 」になる）。

【 0 0 2 8 】

B M 2 ビットマップは、更新プロセスの間に、マスタ・ディスクがローカル・ディスクにすでにコピーされたか否かを示し、設計の都合から、コピーされたことをビット値「 0 」により示す。つまり、B M 2 [D N] [y] が「 0 」であれば、すでにマスタ・ディスクは、ローカル・ディスク [D N] のブロック y にコピーされていることを示す。これらのビットマップの用法については、本発明のオペレーションの説明の中で述べる。

10

【 0 0 2 9 】

図 2、3、および 5 ~ 9 に戻り、本発明の一実施態様に従った更新オペレーションについて考察する。システムのブート・アップ時に、ストレージ・デバイス 0 1 0 8 0 1 内のすべてのディスクのコンテンツが同期化される；すなわち、マスタ・ディスクおよび 2 つのローカル・ディスクの有する情報がまったく同一になる。好ましくは、ディスク・ドライブを互いの鏡像とする。システムのブート・アップ時において、各ウェブ・サーバ・マシンは、ローカル・ディスク [2] をいずれかのディレクトリにマウントする。R o o t D N の値は 2 であり、したがって各ウェブ・サーバ・マシンは、ローカル・ディスク [2] のコンテンツを読み出すことになる。また各ウェブ・サーバ・マシンは、R e q N [1] を 0 に、かつ R e q N [2] を 0 にそれぞれ初期化する。各ウェブ・サーバ・マシンは、独自の H T T P デモン（図 9）および独自の R e M o u n t（リマウント）デモン（図 2 のステップ 0 2 0 3 0 1）を起動する。オーサリング・サーバは、D N を 1 に初期化し、それによってローカル・ディスク [1] がいずれのウェブ・サーバ・マシンによってもマウントされていないことが示される。図 3 は、このプロセスの間におけるシステム・コンポーネントのインタラクションを示している。

20

【 0 0 3 0 】

図 6 を参照すると、システムのブート・アップ時において、ディスク・コントローラ 0 1 0 9 0 1 は、以下の処理を開始する。

【 0 0 3 1 】

ステップ 0 6 0 1 0 2 :

30

ディスク・コントローラがすべてのビットマップを初期化する。つまり、すべての x および y に関して、B M 1 [x] [y] = B M 2 [x] [y] = 0 とする。

【 0 0 3 2 】

ステップ 0 6 0 1 0 3 :

ディスク・コントローラは、ウェブ・サーバ・マシンの 1 つからのメッセージを待機する。メッセージを受け取ると、ディスク・コントローラはステップ 0 6 0 1 0 4 に進む。

【 0 0 3 3 】

ステップ 0 6 0 1 0 4 :

受け取ったメッセージが R e a d M S G（読み出しメッセージ）であれば、ディスク・コントローラは R E A D（読み出し）プロシージャ（図 7）に進む。本発明の一実施態様においては、たとえば R e a d M S G（読み出しメッセージ）を S C S I 標準の読み出しコマンドとする。

40

【 0 0 3 4 】

ステップ 0 6 0 1 0 5 :

受け取ったメッセージが W r i t e M S G（書き込みメッセージ）であれば、ディスク・コントローラは W R I T E（書き込み）プロシージャ（図 8）に進む。本発明の一実施態様においては、たとえば W r i t e M S G（書き込みメッセージ）を S C S I 標準の書き込みコマンドとする。

【 0 0 3 5 】

ステップ 0 6 0 1 0 6 :

50

受け取ったメッセージが `ResyncMSG` (再同期メッセージ) であれば、ディスク・コントローラは `RESYNC INIT` (再同期起動) プロシージャ (図5) に進む。

【0036】

ディスク・コントローラは、以上のステップを連続して繰り返す。

【0037】

図3および9を参照するが、各HTTPデーモンは、ブラウズ中のクライアント010101、010102、...、010103からの要求を処理する。図3は、このデータ・フローを丸囲みの数字1を用いてラベル付けしたラインにより図式的に示している。HTTPデーモン (HTTPd) の一般的なフローは、次のようになる。

【0038】

10

ステップ090102:

HTTPdが、ブラウズ中のクライアントから要求を受け取る。

【0039】

ステップ090103:

HTTPdは、受け取った要求からファイル名を獲得する。たとえば、要求されたURLが「`http://www.foo.com/index.html`」であった場合、HTTPdは、このURLからファイル名「`index.html`」を抽出する。

【0040】

ステップ090104:

一時変数 `y` に `RootDN` の値をストアする。

20

【0041】

ステップ090105:

HTTPdは、`ReqN[y]` を1インクリメントして、ローカル・ディスク [`y`] に対する未決定のアクセスが1つ増えたこと、つまりローカル・ディスク [`y`] からのファイル「`index.html`」の読み出しが増えたことを示す。このステップは、アトミック・プロセスである。つまりこれは、このステップの間、それを除く他のプロセスが `ReqN[y]` にアクセスできないことを意味する。

【0042】

ステップ090106:

HTTPdは、要求されたファイルのコンテンツをローカル・ディスク [`y`] から読み出し、ファイルの一部もしくは全部を、要求を発しているクライアントに送信する。各ウェブ・サーバのファイル・システムは、それぞれのメモリ内にこのファイルをキャッシュすることができる。要求されたファイルがすでにキャッシュされている場合には、HTTPdは、キャッシュ・メモリからそのファイルを読み出す。キャッシュされたファイルは、そのファイルをストアしているディレクトリがアンマウントされると、無効化されるかフラッシュされる。

30

【0043】

ステップ090107:

HTTPdは、`ReqN[y]` を1デクリメントして、ローカル・ディスク [`y`] に対する未決定の要求が、この時点において1つ少なくなったことを示す。このステップもまた、アトミック・プロセスである。なお、ステップ090106が完了するまで、このステップが実行されないことに注意されたい。

40

【0044】

HTTPデーモンは、システムによって受け取られる、データを求めるクライアントの要求のそれぞれに関してこれらのステップを繰り返す。ここで、各ウェブ・サーバ・マシンが、それぞれのシステム上で複数のHTTPデーモンを走らせることができる点に注意する。さらに、ウェブ・サーバ・マシンが、要求されたファイルにアクセスし、コンテンツをクライアントに送信することによってクライアントの要求を満たしているとき、オーサリング・サーバによるマスタ・ディスクのコンテンツに対する更新/追加が可能な点にも注意を要する。

50

【 0 0 4 5 】

図 2 を参照すると、システムのブート・アップ時に、各ウェブ・サーバ・マシン内において R e M o u n t (リマウント) デモンが開始して待機状態 0 2 0 3 0 2 に入ることがわかる。この待機状態においては、R e M o u n t (リマウント) デモンが、オーサリング・サーバ 0 1 0 4 0 1 から送られる R e M o u n t - M S G (リマウント・メッセージ) を待機している。

【 0 0 4 6 】

ここで図 2 および 3 を参照するが、前述したように、システムのオペレーションが正常に行なわれている間は、オーサリング・サーバが、マスタ・ディスク内においてコンテンツの作成および/または更新を行う。図 3 には、このアクティビティが丸囲みの数字 2 を用いてラベル付けした矢印で示されたデータ・フローにより示されている。このアクティビティの結果として、ローカル・ディスクがマスタ・ディスクとの同期から外れる。したがって、ローカル・ディスクの更新が必要になる。このプロセスは、ミラー更新と呼ばれる。図 2 に、このプロセスの概略のステップを示した。

10

【 0 0 4 7 】

同期は、コンテンツ管理者によって、あるいはスケジュールに従って、またはその他のスケジュール基準に基づいてトリガすることができる。更新が望まれたとき、ミラー更新のプロセスが開始されて (ステップ 0 2 0 2 0 1)、次に示す処理に進む。

【 0 0 4 8 】

ステップ 0 2 0 2 0 2 :

20

オーサリング・サーバが、マスタ・ディスク内のコンテンツの作成および更新を停止する。

【 0 0 4 9 】

ステップ 0 2 0 2 0 3 :

オーサリング・サーバのメモリ内にキャッシュされているコンテンツが、マスタ・ディスクに書き込まれた後、フラッシュされる。

【 0 0 5 0 】

ステップ 0 2 0 2 0 4 :

オーサリング・サーバからディスク・コントローラに、ローカル・ディスク [D N] の同期を行うことを示す表示とともに、R e s y n c M S G (再同期メッセージ) が送られる。図 3 においては、これが、丸囲みの数字 3 を用いてラベル付けしたラインによって示されている。このメッセージによってディスク・コントローラは、マスタ・ディスク内の更新されたディスク・ブロックをローカル・ディスク [D N] にコピーする。ここで、いずれのウェブ・サーバ・マシンもローカル・ディスク [D N] をマウントしていないことに注意する。

30

【 0 0 5 1 】

ステップ 0 2 0 2 0 5 :

その後オーサリング・サーバは、待機状態に入り、ディスク・コントローラから C o m p l e t e M S G (完了メッセージ) を受け取るまで休止を設定する。

【 0 0 5 2 】

40

ステップ 0 6 0 1 0 8 :

R e s y n c M S G (再同期メッセージ) がディスク・コントローラに送られる (図 6)。

【 0 0 5 3 】

ステップ 0 6 0 1 0 3 (図 3 の丸囲みの数字 4) :

図 6 を参照すると、ディスク・コントローラが R e s y n c M S G (再同期メッセージ) を受け取る。プロセスは、R E S Y N C I N I T (再同期起動) (図 2 のステップ 0 2 0 1 0 1) に継続される。ディスク・コントローラは、オーサリング・サーバによって送られた R e s y n c M S G (再同期メッセージ) から D N を受け取る。これは、ディスク・コントローラに対して、マスタ・ディスクから更新されたディスク・ブロックをローカ

50

ル・ディスク [D N] にコピーする指示を与える。これより前の再同期デーモンが処理中である場合には、ミラー更新プロシージャが、その再同期デーモンが完了するまで待機することも可能であり、あるいはディスク・コントローラからオーサリング・サーバに通知を行い、後の適切な時期にオーサリング・サーバにリトライさせることも可能である。これらは、設計上の選択肢となる。

【 0 0 5 4 】

ステップ 0 2 0 1 0 2 :

すべての x について、B M 2 [D N] [x] を B M 2 [D N] [x] と B M 1 [D N] [x] の間の論理和にセットする。

【 0 0 5 5 】

ステップ 0 2 0 1 0 3 :

すべての x について、B M 1 [D N] [x] をゼロにセットする。

【 0 0 5 6 】

ステップ 0 2 0 1 0 4 :

ディスク・コントローラが、C o m p l e t e M S G (完了メッセージ) をオーサリング・サーバに送る。

【 0 0 5 7 】

ステップ 0 2 0 1 0 5 :

ディスク・コントローラは、再同期デーモン (R e s y n c d) を起動し、マスタ・ディスク内の新しいコンテンツを用いてローカル・ディスク [D N] を更新する。これにより、プロセスが R E S Y N C D S T A R T (再同期デーモン開始) (図 5 のステップ 0 5 0 1 0 1) に進む。R e s y n c d (再同期デーモン) のプロシージャについては、図 5 と関連から後述する。

【 0 0 5 8 】

ステップ 0 2 0 1 0 6 :

最終的に再同期デーモンが完了する。この後、ディスク・コントローラは、再度メッセージの待機状態に入る (図 6 のステップ 0 6 0 1 0 8) 。

【 0 0 5 9 】

ステップ 0 2 0 2 0 5 :

ここで、オーサリング・サーバが、ディスク・コントローラからの C o m p l e t e M S G (完了メッセージ) を待機していることを思い出されたい。オーサリング・サーバは、それを受け取ると、ステップ 0 2 0 2 0 6 に進む。

【 0 0 6 0 】

ステップ 0 2 0 2 0 6 (図 3 の丸囲みの数字 2) :

本発明によれば、オーサリング・サーバは、更新プロセスが完了していなくてもマスタ・ディスク内のコンテンツの作成および更新を開始する。このアプローチは、オーサリング・サーバが更新したばかりのブロックを修正することはほとんどないという原理に基づいて採用されている。しかしながら、オーサリング・サーバが更新したばかりのブロックの修正を希望することもあり、その場合には、図 8 に示した W r i t e (書き込み) ルーチンによってこの問題が解決される。

【 0 0 6 1 】

ステップ 0 2 0 2 0 7 (図 3 の丸囲みの数字 5) :

オーサリング・サーバからすべてのウェブ・サーバ・マシン上の R e M o u n t (リマウント) デーモンに対して、R e M o u n t M S G (リマウント・メッセージ) が D N 表示とともに送られる。このメッセージによってウェブ・サーバ・マシンは、続いてクライアントから受け取る読み出し要求を、最新の情報を収容しているローカル・ディスク [D N] から満たすことになる。

【 0 0 6 2 】

ステップ 0 2 0 3 0 2 :

各 R e M o u n t (リマウント) デーモンが、オーサリング・サーバから R e M o u n t

10

20

30

40

50

MSG (リマウント・メッセージ) を受け取る。

【0063】

ステップ020303 (図3の丸囲みの数字6) :

各Remount (リマウント) デーモンが、ローカル・ディスク [DN] をマウントする。RootDN = DN をセットする。その後、HTTP デーモンがローカル・ディスク [DN] からデータを読み出す。

【0064】

ステップ020304 :

ReqN [(DN + 1) mod 2] = 0 であれば、各Remount (リマウント) デーモンは、ローカル・ディスク [(DN + 1) mod 2] をアンマウントすることができる。それに該当しなければ、Remount (リマウント) デーモンは、ReqN [(DN + 1) mod 2] が「0」になるまで待機する。これにおいて、「DN」は、マスタ・ディスクを用いて更新されたディスクを表し、「(DN + 1) mod 2」は、更新されなかったディスクを表す。

10

【0065】

ステップ020305 (図3の丸囲みの数字7) :

各Remount (リマウント) デーモンが、ローカル・ディスク [(DN + 1) mod 2] をアンマウントする。

【0066】

ステップ020306 (図3の丸囲みの数字8) :

各Remount (リマウント) デーモンが、CompleteMSG (完了メッセージ) をオーサリング・サーバに送る。

20

【0067】

ステップ020208 :

オーサリング・サーバは、ウェブ・サーバ・マシンのそれぞれからCompleteMSG (完了メッセージ) を受け取るまで待機している。

【0068】

ステップ020209 :

すべてのウェブ・サーバ・マシンが更新されなかったディスクをアンマウントすれば、DNの更新が可能になる。つまり、DN = (DN + 1) mod 2。

30

【0069】

ステップ020210 :

MIRROR UPDATE (ミラー更新) のプロセスを終了する。

【0070】

図2および5を参照すると、ステップ020105において、ディスク・コントローラが再同期デモンを起動している。このデモンは、更新されたディスク・ブロックをマスタ・ディスクからDNによって識別されるローカル・ディスクにコピーする。図5に示したフローチャートは、再同期デモンのプロセスの概要を表している。

【0071】

ステップ050102 :

一時変数DBNを「1」にセットする。なお、このプロシージャにおいて使用する一時変数xは、再同期デモンによる、マスタ・ディスクからローカル・ディスク[x]への更新されたディスク・ブロックのコピーを表していることに注意されたい。xの値は、DNにセットされる。

40

【0072】

ステップ050103 :

DBNがディスク・ブロックの最大数より大きい場合には、再同期デモンが終了する (ステップ050108)。

【0073】

ステップ050104 :

50

マスタ・ディスク内のディスク・ブロック [D B N] は更新されているが、コピーが行なわれていない場合には、ステップ 0 5 0 1 0 5 に進む。それに該当しない場合は、ステップ 0 5 0 1 0 7 に進む。ここで、 $B M 2 [x] [D B N] = 1$ は、マスタ・ディスク内のディスク・ブロック [D B N] がローカル・ディスクにまだコピーされていないことを表す点に注意が必要である。

【 0 0 7 4 】

ステップ 0 5 0 1 0 5 :

マスタ・ディスク内のディスク・ブロック [D B N] をローカル・ディスク [x] 内のディスク・ブロック [D B N] にコピーする。これからわかるように、本発明の特徴の 1 つは、ディスクからディスクへのデータの直接転送である。この転送は、ストレージ・デバイス 0 1 0 8 0 1 内のデータ・パスに沿って行なわれ、それにはデータ・スイッチ 0 1 0 6 0 1 が含まれない。

10

【 0 0 7 5 】

ステップ 0 5 0 1 0 6 :

このプロセスがローカル・ディスク [x] のディスク・ブロック [D B N] を更新する。 $B M 2 [x] [D B N]$ を「 0 」にセットして、そのブロックがコピーされたことを示す。

【 0 0 7 6 】

ステップ 0 5 0 1 0 7 :

$D B N = D B N + 1$ 、すなわち次のブロックへのインクリメントを行う。上記のプロシー

20

【 0 0 7 7 】

ステップ 0 5 0 1 0 8 :

$R e s y n c d$ (再同期デーモン) のプロセスを完了する。

【 0 0 7 8 】

図 7 を参照するが、ローカル・ディスク [x] 内のディスク・ブロックの更新は、再同期デーモンが開始した後に即座に行なわれるわけではない。更新されたブロックをマスタ・ディスクからローカル・ディスクにコピーするためには、有限の所定時間を要する。一方、その間にこれらのディスク・ブロックの 1 つに対する読み出しアクセスがあった場合には、ディスク・コントローラから要求発信側に対して、ローカル・ディスクが完全に更新されているという印象が与えられなければならない。ディスク・コントローラは、ローカル・ディスクに更新済みの情報が含まれているか否かを判断しなければならず、含まれていなければ、要求されたディスク・ブロックをローカル・ディスク [x] からではなく、マスタ・ディスクから読み出さなければならない。図 7 のフローチャートは、この状況における読み出しプロシーダの概要を示している。

30

【 0 0 7 9 】

ステップ 0 7 0 1 0 1 :

ディスク・コントローラが、 $R e a d M S G$ (読み出しメッセージ) を受け取ると、ステップ 0 6 0 1 0 4 からこのプロシーダが呼び出される。一時変数 $D B N$ および x は、要求発信側が、ローカル・ディスク [x] 内のディスク・ブロック [D B N] の読み出しを試みていることを示す。

40

【 0 0 8 0 】

ステップ 0 7 0 1 0 2 :

$R e a d M S G$ (読み出しメッセージ) が、要求発信側がマスタ・ディスクからのデータの読み取りを試みていること (オーサリング・サーバの場合等) を示している場合には、マスタ・ディスクが、その定義から最新のコンテンツを含んでいることから、ディスク・コントローラは、単純にステップ 0 7 0 1 0 8 に進む。読み出し要求がローカル・ディスクに関するものであれば、ステップ 0 7 0 1 0 3 に進む。

【 0 0 8 1 】

ステップ 0 7 0 1 0 3 :

50

ローカル・ディスク [x] 内のディスク・ブロック [D B N] がすでに更新されていれば (つまり、B M 2 [x] [D B N] = 0)、ステップ 0 7 0 1 0 4 に進む。更新されていなければ、ステップ 0 7 0 1 0 5 に進む。

【 0 0 8 2 】

ステップ 0 7 0 1 0 4 :

ディスク・コントローラは、ローカル・ディスク [x] からディスク・ブロック [D B N] を読み出し、そのディスク・ブロックを要求発信側に送る。R E A D (読み出し) プロシージャを終了する (ステップ 0 7 0 1 0 9)。

【 0 0 8 3 】

ステップ 0 7 0 1 0 5 :

ディスク・コントローラは、マスタ・ディスクからディスク・ブロック [D B N] を読み出し、そのディスク・ブロックを要求発信側に送る。

【 0 0 8 4 】

ステップ 0 7 0 1 0 6 :

ディスク・コントローラは、ステップ 0 7 0 1 0 5 において読み込んだディスク・ブロック [D B N] をローカル・ディスク [x] 内のディスク・ブロック [D B N] に書き込む。このプロセスは、ローカル・ディスク [x] 内のディスク・ブロック [D B N] を更新することになる。

【 0 0 8 5 】

ステップ 0 7 0 1 0 7 :

ローカル・ディスク [x] 内のディスク・ブロック [D B N] が更新されたことから、B M 2 [x] [D B N] を「 0 」にセットする。R E A D (読み出し) プロシージャを終了する (ステップ 0 7 0 1 0 9)。

【 0 0 8 6 】

ステップ 0 7 0 1 0 8 :

ディスク・コントローラは、マスタ・ディスクからディスク・ブロック [D B N] を読み出し、そのディスク・ブロックを要求発信側に送る。R E A D (読み出し) プロシージャを終了する (ステップ 0 7 0 1 0 9)。

【 0 0 8 7 】

ステップ 0 7 0 1 0 3、0 7 0 1 0 5、0 7 0 1 0 6、および 0 7 0 1 0 7 からなるパスは、クリティカル・パスを構成し、その重要性については説明を後述する。

【 0 0 8 8 】

図 8 を参照するが、ローカル・ディスク [x] 内のディスク・ブロックの更新は、再同期デーモンが開始した後に即座に行なわれるわけではない。更新されたブロックをマスタ・ディスクからローカル・ディスクにコピーするためには、有限の所定時間を要する。その間にオーサリング・サーバは、次の作成 / 更新アクティビティ (図 2 のステップ 0 2 0 2 0 6) のラウンドに進む。その結果、マスタ・ディスクのブロックがローカル・ディスクにコピーされる前に、それに対する書き込みが試みられる可能性がある。図 8 のフローチャートは、この状況を扱うプロシージャの概要を示している。

【 0 0 8 9 】

ステップ 0 8 0 1 0 1 :

ディスク・コントローラが W r i t e M S G (書き込みメッセージ) を受け取ると、このプロシージャがステップ 0 6 0 1 0 5 から呼び出される。一時変数 D B N は、要求発信側 (つまり、オーサリング・サーバ) が、マスタ・ディスク内のディスク・ブロック [D B N] に対する書き込みを希望していることを示す。なお、ウェブ・サーバ・マシンは、ローカル・ディスクに対して読み出し要求のみを発することから、ローカル・ディスクに対する書き込みアクセスがないことに注意する必要がある。

【 0 0 9 0 】

ステップ 0 8 0 1 0 2 :

一時変数 x は、ローカル・ディスクをインデクスするために使用される。x = 1 にセット

10

20

30

40

50

する。

【 0 0 9 1 】

ステップ 0 8 0 1 0 3 :

x > N D I S K であれば、ステップ 0 8 0 1 0 9 に進む。それに該当しなければ、ステップ 0 8 0 1 0 4 に進む。ただし、これにおいて N D I S K は、ローカル・ディスクの数、つまり 2 である。

【 0 0 9 2 】

ステップ 0 8 0 1 0 4 :

ローカル・ディスク [x] 内のディスク・ブロック [D B N] がまだ更新されていなければ (つまり B M 2 [x] [D B N] = 1)、マスタ・ディスク内の対応するディスク・ブロックを上書きする前に、そのディスク・ブロックを更新しなければならない。その場合には、ステップ 0 8 0 1 0 5 に進む。ローカル・ディスク [x] 内のディスク・ブロック [D B N] がすでに更新されていれば、ステップ 0 8 0 1 0 7 に進む。

10

【 0 0 9 3 】

ステップ 0 8 0 1 0 5 :

ディスク・コントローラがマスタ・ディスク内のディスク・ブロック [D B N] を、ローカル・ディスク [x] 内のディスク・ブロック [D B N] に書き込む。

【 0 0 9 4 】

ステップ 0 8 0 1 0 6 :

ローカル・ディスク [x] 内のディスク・ブロック [D B N] が更新されたことから、B M 2 [x] [D B N] を「 0 」にセットして、この事実を反映させる。

20

【 0 0 9 5 】

ステップ 0 8 0 1 0 7 :

ステップ 0 8 0 1 0 9 において、ディスク・コントローラがマスタ・ディスク内のディスク・ブロック [D B N] に書き込みを行い、その結果このブロックが更新されることになる。オーサリング・サーバから R e s y n c M S G (再同期メッセージ) が送られたときには (図 2 のステップ 0 2 0 2 0 4)、ディスク・コントローラが、このディスク・ブロックをすべてのローカル・ディスクにコピーしなければならない。これは、B M 1 [x] [D B N] を「 1 」にセットし、ブロック D B N が新しい情報を含んでいることを示すことによって達成される。

30

【 0 0 9 6 】

ステップ 0 8 0 1 0 8 :

すべてのローカル・ディスクに関して、上記のプロセス (ステップ 0 8 0 1 0 3 からステップ 0 8 0 1 0 8 まで) を繰り返す。

【 0 0 9 7 】

ステップ 0 8 0 1 0 9 :

ここで始めてディスク・コントローラは、W r i t e M S G (書き込みメッセージ) のプロセスに進むことができる。ディスク・コントローラは、マスタ・ディスク内のディスク・ブロック [D B N] に書き込みを行う。W R I T E (書き込み) プロシージャを終了する (ステップ 0 8 0 1 1 0)。

40

【 0 0 9 8 】

ステップ 0 8 0 1 0 4、0 8 0 1 0 5、および 0 8 0 1 0 6 からなるステップは、クリティカル・パスを構成し、その重要性については説明を後述する。

【 0 0 9 9 】

図 4 のシステム・ブロック図および図 1 4 のフローチャートを参照し、本発明のもう 1 つの実施態様について考察する。この場合、図 4 に示したデータ・ストレージ・システム 0 1 0 8 0 1 は、2 つのローカル・ディスク 0 1 1 0 0 1、0 1 1 1 0 2 を包含している。この実施態様においては、マスタ・ディスクが存在しない。オーサリング・サーバは、ローカル・ディスクの一方においてコンテンツの作成および更新を行う。そのほかのコンポーネントは、図 2 に示した実施態様と同じになる。

50

【0100】

ここで簡単に、図10、11、18、および19を参照すると、各種の内部データ変数が上記のコンポーネントによって維持されていることがわかる。オーサリング・サーバ010401は、アンマウント・ディスク番号DN 160101（図13）を管理している。このデータ変数は、ローカル・ディスク[DN]が、いずれのウェブ・サーバ・マシンによってもマウントされていないことを示す。

【0101】

各ウェブ・サーバ・マシン010501、...、01050Nは、そのルート・ディレクトリのディスク番号、RootDN 110101（図11）を管理している。この変数は、各ウェブ・サーバ・マシンが、ローカル・ディスク[RootDN]からコンテンツの読み出しを行っていることを示す。また各ウェブ・サーバ・マシンは、各ローカル・ディスクに関する完了していない要求の数を管理する。たとえば、ReqN[1] 100101（図10）は、ローカル・ディスク[1]に関する完了していない要求の数を表す。ウェブ・サーバ・マシンは、そのReqN[i]がゼロである場合に限ってローカル・ディスク[i]をアンマウントすることができる。

10

【0102】

ストレージ・デバイス010801は、ビットマップのセットBM1[x] 210301、およびBM2[x] 210401（図18）を管理している。MaxDBNは、ローカル・ディスクのディスク・ブロックの最大数である。本発明においては、すべてのディスク・ドライブの有するディスク・ブロックが同数となる。変数PARENT 220101およびCHILD 220102（図19）も提供されている。これらの変数PARENT（親）およびCHILD（子）は、ローカル・ディスクの番号1または2をストアする。慣習に従ってローカル・ディスク[PARENT]は、ローカル・ディスク[CHILD]より新しいコンテンツを有する。つまり、再同期デーモンは、更新されたディスク・ブロックをローカル・ディスク[PARENT]からローカル・ディスク[CHILD]にコピーする。

20

【0103】

図6、9、14~17、および20を参照すると、システムのブート・アップ時に、各ウェブ・サーバ・マシン010501、...、01050Nは、ローカル・ディスク[2]をルート・ディレクトリとしてマウントする。RootDNの値は2であり、したがってウェブ・サーバは、ローカル・ディスク[2]からコンテンツを読み出す。また各ウェブ・サーバは、ReqN[1]=0およびReqN[2]=0に初期化する。各ウェブ・サーバ・マシンは、それぞれ独自のHTTPデーモン（図9のステップ090101）および独自のRemount（リマウント）デーモン（図14のステップ170301）を起動する。HTTPデーモンは、ブラウザ中のクライアントからの要求を処理する。Remount（リマウント）デーモンは、オーサリング・サーバからのRemount-MSG（リマウント・メッセージ）を待機する（ステップ170302）。

30

【0104】

オーサリング・サーバは、DNを「1」に初期化し、いずれのウェブ・サーバ・マシンもローカル・ディスク[1]をマウントしていないことを示す。システムのブート・アップ時には、ローカル・ディスク[1]およびローカル・ディスク[2]が同一のコンテンツを含んでいる。またシステムのブート・アップ時において、ディスク・コントローラは、図6を参照して前述した方法に従って処理を開始する。本発明のこの実施態様においては、ディスク・コントローラが、すべてのxについて初期化を行い、BM1[x]およびBM2[x]が「0」にセットされる。

40

【0105】

本発明のこの実施態様におけるディスク・コントローラのプロセス（図6）およびHTTPデーモンのプロセス（図9）は、前述の説明と同じになる。

【0106】

図14を参照すると、本発明のこの実施態様に従った更新プロセスのフローチャートが示

50

されている。

【0107】

ステップ170202:

オーサリング・サーバが、ローカル・ディスク[DN]内のコンテンツの作成および更新を停止する。

【0108】

ステップ170203:

オーサリング・サーバがメモリ内にキャッシュしているデータが、ローカル・ディスク[DN]にフラッシュされる。

【0109】

ステップ170204(図20の丸囲みの数字3):

オーサリング・サーバから、すべてのウェブ・サーバ・マシン上のReMount(リマウント)デーモンに対して、ReMountMSG(リマウント・メッセージ)が送られる。このメッセージによってウェブ・サーバ・マシンは、続くクライアントからのデータを求める要求を満たすために、ローカル・ディスク[DN]からコンテンツを読み出すことになる。

【0110】

ステップ170302:

各ReMount(リマウント)デーモンが、オーサリング・サーバからReMountMSG(リマウント・メッセージ)を受け取る。

【0111】

ステップ170303(図20の丸囲みの数字4):

各ReMount(リマウント)デーモンが、そのルート・ディレクトリとしてローカル・ディスク[DN]をマウントする。RootDN=DNをセットする。この時点以降は、データを求めるクライアントの要求を満たすために、ローカル・ディスク[DN]から読み出したデータが用いられる。

【0112】

ステップ170304:

$ReqN[(DN+1) \bmod 2] = 0$ であれば、各ReMount(リマウント)デーモンは、ローカル・ディスク $[(DN+1) \bmod 2]$ をアンマウントすることができる。それに該当しなければ、ReMount(リマウント)デーモンは、 $ReqN[(DN+1) \bmod 2]$ が「0」になるまで待機する。

【0113】

ステップ170305(図20の丸囲みの数字5):

各ReMount(リマウント)デーモンが、ローカル・ディスク $[(DN+1) \bmod 2]$ をアンマウントする。

【0114】

ステップ170306(図20の丸囲みの数字6):

各ReMount(リマウント)デーモンは、以前のディスク(つまり、 $(DN+1) \bmod 2$)のアンマウントに成功すると、CompleteMSG(完了メッセージ)をオーサリング・サーバに送る。

【0115】

ステップ170205:

オーサリング・サーバは、すべてのウェブ・サーバ・マシンからCompleteMSG(完了メッセージ)を受け取るまで待機しており、それをすべて受け取った時点においては、以前のディスクをマウントしているウェブ・サーバ・マシンがなくなったことになる。

【0116】

ステップ170206(図20の丸囲みの数字7):

オーサリング・サーバが、ディスク・コントローラにResyncMSG(再同期メッセ

10

20

30

40

50

ージ)を送る。このメッセージによってディスク・コントローラは、ローカル・ディスク [DN] 内の更新されたディスク・ブロックをローカル・ディスク [(DN + 1) mod 2] にコピーする。この時点においては、いずれのウェブ・サーバ・マシンもローカル・ディスク [(DN + 1) mod 2] をマウントしていないことに注意を要する。

【0117】

ステップ060108:

ResyncMSG (再同期メッセージ) がディスク・コントローラに送られる。

【0118】

ステップ060103 (図20の丸囲みの数字8):

ディスク・コントローラは、ResyncMSG (再同期メッセージ) を受け取り、RESYNC INIT (再同期起動) (図14のステップ170101) に進む。この時点において、ディスク・コントローラは、オーサリング・サーバによって送られたResyncMSG (再同期メッセージ) からDNを受け取る。これは、ディスク・コントローラに対して、更新されたディスク・ブロックをローカル・ディスク [DN] からローカル・ディスク [(DN + 1) mod 2] にコピーすることを指示する。

10

【0119】

ステップ170102:

これより以前の再同期デーモンが処理中である場合には、ミラー更新プロシージャが、その再同期デーモンが完了するまで待機する。

【0120】

20

ステップ170103:

変数PARENT (親) およびCHILD (子) の定義から、PARENT = DNをセットする。

【0121】

ステップ170104:

CHILD = (DN + 1) mod 2をセットする。

【0122】

ステップ170105:

すべてのxについて、BM2[x] = BM1[x]をセットする。

【0123】

30

ステップ170106:

すべてのxについて、BM1[x] = 0をセットする。

【0124】

ステップ170107:

ディスク・コントローラが、CompleteMSG (完了メッセージ) をオーサリング・サーバに送る。

【0125】

ステップ170108:

ディスク・コントローラは、再同期デーモンを実行する (図15のステップ180101のRESYNCD START (再同期デーモン開始))。このデーモン・プロセスは、更新を実現するコピー・オペレーションを実際に実行するプロセスである。

40

【0126】

ステップ170109:

RESYNC INIT (再同期起動) のプロセスを終了する。その後ディスク・コントローラは、別のメッセージの待機状態に入る (図6)。

【0127】

ステップ170207:

オーサリング・サーバは、ディスク・コントローラからのCompleteMSG (完了メッセージ) を待機している。オーサリング・サーバは、それを受け取ると、ステップ170208に進む。

50

【 0 1 2 8 】

ステップ 1 7 0 2 0 8 :

オーサリング・サーバは、ローカル・ディスク $[(DN + 1) \bmod 2]$ 内において、次の作成および/または更新アクティビティのラウンドを開始する。この時点においては、再同期デーモン・プロセスが更新の実行を処理中であることに注意する。

【 0 1 2 9 】

ステップ 1 7 0 2 0 9 :

すべてのウェブ・サーバ・マシンが、すでにローカル・ディスク $[(DN + 1) \bmod 2]$ をアンマウントしていることから、DN が更新される ; $DN = (DN + 1) \bmod 2$ 。

10

【 0 1 3 0 】

ステップ 1 7 0 2 1 0 :

ミラー更新のプロセスを終了する。

【 0 1 3 1 】

次に、図 1 5 を参照して本発明のこの実施態様における再同期デーモンについて説明する。ここで、ステップ 1 7 0 1 0 8 において、ディスク・コントローラが再同期デーモン・プロセスを実行していることを思い出されたい。このデーモン・プロセスは、ローカル・ディスク $[PARENT]$ 内の更新されたディスク・ブロックをローカル・ディスク $[CHILD]$ にコピーする。以下のフローチャートは、このプロシージャの概要を表している。

20

【 0 1 3 2 】

ステップ 1 8 0 1 0 2 :

一時変数 DBN を「 1 」にセットする。

【 0 1 3 3 】

ステップ 1 8 0 1 0 3 :

DBN がディスク・ブロックの最大数 $(MaxDBN)$ より大きい場合には、再同期デーモンが終了する (ステップ 1 8 0 1 0 8)。

【 0 1 3 4 】

ステップ 1 8 0 1 0 4 :

ローカル・ディスク $[CHILD]$ 内のディスク・ブロック $[DBN]$ が更新されている場合には、プロセスがステップ 1 8 0 1 0 5 に進む。それに該当しなければ、プロセスがステップ 1 8 0 1 0 7 に進む。なお、 $BM2[DBN] = 1$ は、ローカル・ディスク $[PARENT]$ 内のディスク・ブロック $[DBN]$ が更新されていることを表す。

30

【 0 1 3 5 】

ステップ 1 8 0 1 0 5 :

ローカル・ディスク $[PARENT]$ 内のディスク・ブロック $[DBN]$ をローカル・ディスク $[CHILD]$ 内のディスク・ブロック $[DBN]$ にコピーする。ほかの実施態様における場合と同様に、ディスクからディスクへの転送が、完全にストレージ・デバイス 0 1 0 8 0 1 の内側において行なわれ、そのデータ・パスにはデータ・スイッチが含まれない。

40

【 0 1 3 6 】

ステップ 1 8 0 1 0 6 :

ローカル・ディスク $[CHILD]$ 内のディスク・ブロック $[DBN]$ は、ステップ 1 8 0 1 0 5 によって更新されている。ここでは、 $BM2[DBN]$ を「 0 」にセットしてその事実を反映させる。

【 0 1 3 7 】

ステップ 1 8 0 1 0 7 :

$DBN = DBN + 1$ を行う。上記のプロシージャを、マスタ・ディスク内のすべてのディスク・ブロックについて繰り返す。

【 0 1 3 8 】

50

ステップ 180108 :

再同期デモン・プロセスを完了する。

【0139】

ステップ 180104、180105、および 180106 からなるステップは、クリティカル・パスを構成し、その重要性については説明を後述する。

【0140】

図 16 を参照するが、再同期デモンが開始してから更新オペレーションが完了するまで、有限の所定時間を必要とする。しかしながら、その間にも読み出し要求は発せられる。つまり、必ずしもすべてのブロックの更新が完了していないことから、読み出しオペレーションの対象となるローカル・ディスク [CHILD] 内のディスク・ブロックの一部は、更新済みの情報を含まないことがある。これらのディスク・ブロックの 1 つに対する読み出しアクセスがあった場合に、ディスク・コントローラは、ローカル・ディスク [CHILD] からではなく、ローカル・ディスク [PARENT] からそのディスク・ブロックを読み出さなければならない。図 16 は、このシナリオに関する読み出しプロシージャの概要を表している。

10

【0141】

ステップ 190101 :

ディスク・コントローラが、ReadMSG (読み出しメッセージ) を受け取ると、ステップ 060104 (図 6) からこのプロシージャが呼び出される。一時変数 DBN は、要求発信側が、一方のローカル・ディスク内のディスク・ブロック [DBN] に対する読み出しオペレーションを指定したことを示す。

20

【0142】

ステップ 190102 :

ReadMSG (読み出しメッセージ) によって、その読み出し要求がローカル・ディスク [PARENT] からのデータの読み取りであることが示される場合には、定義により PARENT (親) は最新の情報を含んでいることから、ディスク・コントローラは、何も特別なことを行う必要がない。したがって、プロセスがステップ 190108 に進む。読み出し要求が CHILD (子) に対するものであるときには、プロセスがステップ 190103 に継続される。

30

【0143】

ステップ 190103 :

ローカル・ディスク [CHILD] 内のディスク・ブロック [DBN] がすでに更新されていれば、ステップ 070104 に進む。更新されていなければ、ステップ 070105 に進む。

【0144】

ステップ 190104 :

ディスク・コントローラは、ローカル・ディスク [CHILD] 内のディスク・ブロック [DBN] を読み出し、そのディスク・ブロックを要求発信側を送る。READ (読み出し) プロシージャを終了する (ステップ 190109)。

40

【0145】

ステップ 190105 :

ディスク・コントローラは、ローカル・ディスク [PARENT] 内のディスク・ブロック [DBN] を読み出し、そのディスク・ブロックを要求発信側を送る。

【0146】

ステップ 190106 :

またディスク・コントローラは、このディスク・ブロックをローカル・ディスク [CHILD] に書き込む。これによってローカル・ディスク [CHILD] 内のディスク・ブロック [DBN] が更新される。

【0147】

ステップ 190107 :

50

ローカル・ディスク [C H I L D] 内のディスク・ブロック [D B N] が更新されたことから、B M 2 [D B N] を「 0 」にセットしてこの事実を反映させる。R E A D (読み出し) プロシージャを終了する (ステップ 1 9 0 1 0 9)。

【 0 1 4 8 】

ステップ 1 9 0 1 0 8 :

ディスク・コントローラは、ローカル・ディスク [P A R E N T] 内のディスク・ブロック [D B N] を読み出し、そのディスク・ブロックを要求発信側に送る。R E A D (読み出し) プロシージャを終了する (ステップ 1 9 0 1 0 9)。

【 0 1 4 9 】

ステップ 1 9 0 1 0 3、1 9 0 1 0 5、1 9 0 1 0 6、および 1 9 0 1 0 7 からなるパスは、クリティカル・パスを構成し、その重要性については説明を後述する。

【 0 1 5 0 】

ここで、図 1 7 を参照して本発明のこの実施態様に従った W R I T E (書き込み) プロセスについて説明する。

【 0 1 5 1 】

ステップ 2 0 0 1 0 1 :

ディスク・コントローラが W r i t e M S G (書き込みメッセージ) を受け取ると、このプロシージャがステップ 0 6 0 1 0 5 (図 6) から呼び出される。一時変数 D B N は、要求発信側が、ローカル・ディスク [C H I L D] 内のディスク・ブロック [D B N] に対する書き込みを試みていることを示す。なお、いずれのウェブ・サーバ・マシンによっても、ローカル・ディスク [P A R E N T] に対する書き込みアクセスが行なわれないことに注意する必要がある。この点において、ウェブ・サーバ・マシンは、ローカル・ディスク [P A R E N T] からコンテンツの読み出しのみを行う。

【 0 1 5 2 】

ステップ 2 0 0 1 0 2 :

ステップ 0 8 0 1 0 9 においては、ディスク・コントローラが、ローカル・ディスク [C H I L D] 内のディスク・ブロック [D B N] の書き込みを行うことになる。オーサリング・サーバから R e s y n c M S G (再同期メッセージ) が送られたとき、このディスク・ブロックをローカル・ディスク [P A R E N T] にコピーすることがディスク・コントローラにわかっているなければならない。このことは、B M 1 [D B N] を「 1 」にセットすることによって維持される。

【 0 1 5 3 】

ステップ 2 0 0 1 0 3 :

ディスク・コントローラは、ローカル・ディスク [C H I L D] 内のディスク・ブロック [D B N] の書き込みを行うことによって W r i t e M S G (書き込みメッセージ) を処理する。W R I T E (書き込み) プロシージャを終了する (ステップ 2 0 0 1 0 4)。

【 0 1 5 4 】

ここで、W r i t e M S G (書き込みメッセージ) によって指定されたデータのサイズがディスク・ブロックのサイズより小さい場合において、B M 2 [D B N] = 1 であれば、ステップ 2 0 0 1 0 3 に先行して、ローカル・ディスク [P A R E N T] 内のディスク・ブロック [D B N] をローカル・ディスク [C H I L D] にコピーしなければならないことに注意する必要がある。これを行う理由については、例を用いることによってもっとも良好な説明が得られる。システムのブート・アップ時においては、ローカル・ディスク [1] 内のディスク・ブロック [D B N] およびローカル・ディスク [2] 内のディスク・ブロック [D B N] が、同一のデータを有している。ある時点において、ローカル・ディスク [2] 内のディスク・ブロック [D B N] が完全に更新されたとする。その結果、オーサリング・サーバが R e s y n c - M S G (再同期メッセージ) を発行し、ローカル・ディスク [2] 内のディスク・ブロック [D B N] をローカル・ディスク [1] にコピーしなければならないことを示す。オーサリング・サーバは、このコピーが完了する前に、ローカル・ディスク [1] 内のディスク・ブロック [D B N] の一部を更新することが

できる。コピーが完了する前に、BM2[DBN]が「0」にセットされると、ローカル・ディスク[2]内のディスク・ブロック[DBN]のほかの部分のローカル・ディスク[1]にコピーされないことがある。結果的にこれは、ローカル・ディスク[1]とローカル・ディスク[2]の間における不一致を招くことになる。

【0155】

Resyncd(再同期デーモン)がいずれかのコピー・オペレーションを処理しており、かつディスク・コントローラが上に示したクリティカル・パス内にあるときは、少なくとも何らかのロック・メカニズムが必要である。これらのクリティカル・パスは、互いに順序設定されていなければならない。2ないしはそれ以上のクリティカル・パスが、同時に実行されてはならない。これは、ビットマップBM1およびBM2が、Resyncd(再同期デーモン)およびディスク・コントローラそれぞれのための共有リソースであることによる。

10

【0156】

図21は、伝統的なWWW環境を示している。各ウェブ・サーバ・マシンは、それぞれ独自のローカル・ディスクを有している。ミラー更新オペレーションを実行するためには、オーサリング・サーバが、マスタ・ディスクからコンテンツを読み出し、それを各ウェブ・サーバ・マシンに送らなければならない。各ウェブ・サーバ・マシンは、図において丸囲みの数字1を用いてラベル付けした実線によって示されるように、受け取ったコンテンツをそれぞれのローカル・ディスク内にストアすることによってこれに応答する。この種のプロセスは、貴重なCPUリソースを消費する；これらのリソースは、データに関するクライアントの要求を満たすために使用された方が好ましい。つまり、大量のデータの更新を必要とするときには、ライブ更新の実行が実用的でなくなる。たとえば、ビデオ・オン・デマンド・システムにおいては、ビデオ・リリースのインストールまたはアンインストールを行うために、システムを外さなければならない。これは、伝統的な方法における制限である。

20

【0157】

【発明の効果】

これに対して、本発明に従った方法およびシステムは、ウェブ・サーバのCPUパワーを消費する必要がない。ストレージ・デバイスが、マスタ・ディスクからローカル・ディスクにデータのコピーを行う唯一のコンポーネントになる。ウェブ・サーバ・マシンに要求されるタスクは、更新プロセスの間におけるディスクのマウントおよびアンマウントだけである。

30

【0158】

共有ファイル・システムのアプローチと比較すると、本発明においては、各ウェブ・サーバ・マシンが通信オーバーヘッドを伴うことなくファイルにアクセスできることから、ファイルにアクセスするときのファイル・サーバのオーバーヘッドが抑えられる。さらに、データ・スイッチの使用がファイル・サーバのボトルネックを回避する。

【図面の簡単な説明】

【図1】本発明に従ったデータ・サーバ構成の一実施態様を表した概要図である。

【図2】本発明の一実施態様に従った更新プロシージャの間におけるメイン・プロセッシング・ステップの概要を表したハイ・レベルのフローチャートである。

40

【図3】本発明の一実施態様に従った更新プロシージャの前およびその間における論理的なデータの流れを示した説明図である。

【図4】本発明に従ったデータ・サーバ構成の別の実施態様を表した概要図である。

【図5】本発明の一実施態様に従った再同期デーモン・プロセスのメイン・プロセッシング・ステップを表した概略のフローチャートである。

【図6】ディスク・コントローラのメイン・プロセッシング・ステップを表した概略のフローチャートである。

【図7】本発明の一実施態様に従った読み出しプロシージャのメイン・プロセッシング・ステップを表した概略のフローチャートである。

50

【図 8】本発明の一実施態様に従った書き込みプロシージャのメイン・プロセッシング・ステップを表した概略のフローチャートである。

【図 9】HTTP デモン・プロセスのメイン・プロセッシング・ステップを表した概略のフローチャートである。

【図 10】本発明の一実施態様に従って使用されるデータ構造を示した説明図である。

【図 11】本発明の一実施態様に従って使用されるデータ構造を示した説明図である。

【図 12】本発明の一実施態様に従って使用されるデータ構造を示した説明図である。

【図 13】本発明の一実施態様に従って使用されるデータ構造を示した説明図である。

【図 14】本発明の別の実施態様に従った更新プロシージャの間におけるメイン・プロセッシング・ステップの概要を表したハイ・レベルのフローチャートである。

10

【図 15】本発明の別の実施態様に従った再同期デモン・プロセスのメイン・プロセッシング・ステップを表した概略のフローチャートである。

【図 16】本発明の別の実施態様に従った読み出しプロシージャのメイン・プロセッシング・ステップを表した概略のフローチャートである。

【図 17】本発明の別の実施態様に従った書き込みプロシージャのメイン・プロセッシング・ステップを表した概略のフローチャートである。

【図 18】本発明の別の実施態様に従って使用されるデータ構造を示した説明図である。

【図 19】本発明の別の実施態様に従って使用されるデータ構造を示した説明図である。

【図 20】本発明の別の実施態様に従った更新プロシージャの前およびその間における論理的なデータの流れを示した説明図である。

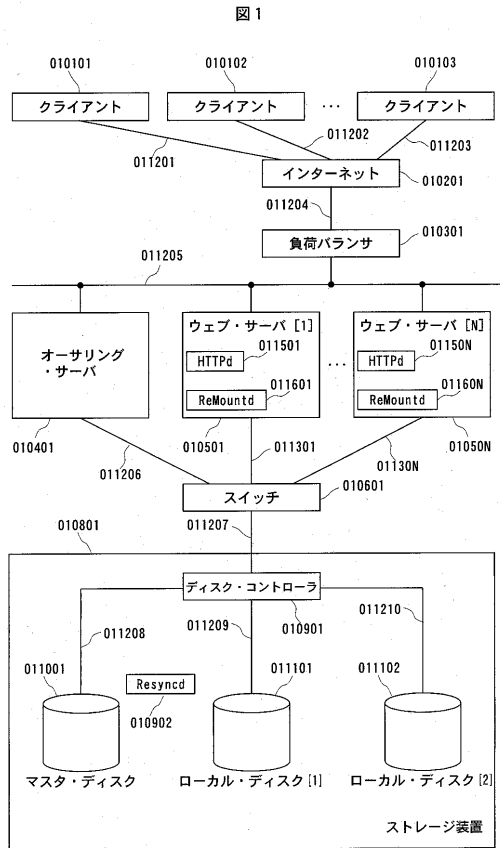
20

【図 21】従来技術構成のデータ・サーバ・システムにおける更新の構成を示した説明図である。

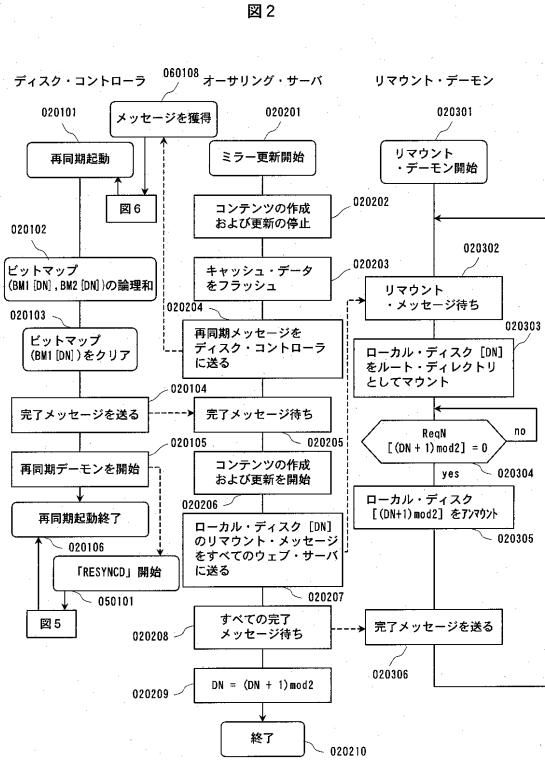
【符号の説明】

0 1 0 1 0 1 : ブラウズ中のクライアント、0 1 0 2 0 1 : インターネット、0 1 0 3 0 1 : 負荷バランサ、0 1 0 4 0 1 : オーサリング・サーバ、0 1 0 5 0 1 : ウェブ・サーバ、0 1 0 6 0 1 : スイッチ、0 1 0 9 0 1 : ディスク・コントローラ、0 1 1 0 0 1 : マスタ・ディスク、0 1 1 1 0 1 : ローカル・ディスク、0 1 0 8 0 1 : ストレージ・デバイス。

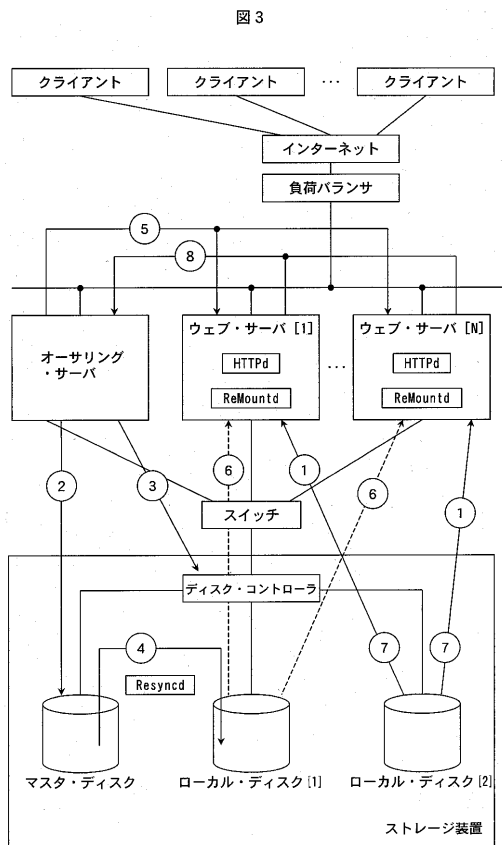
【図 1】



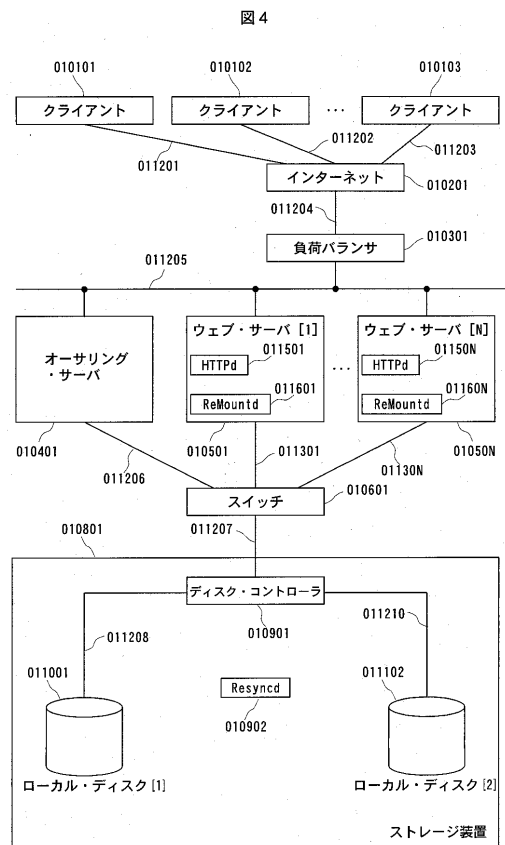
【図 2】



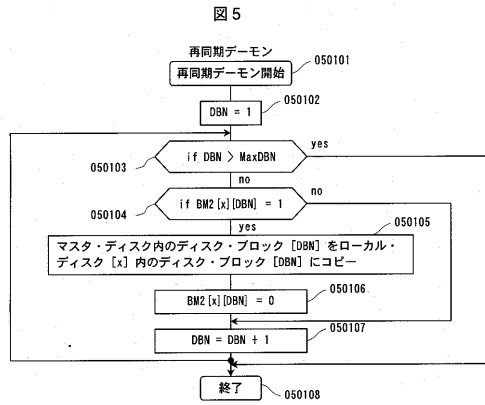
【図 3】



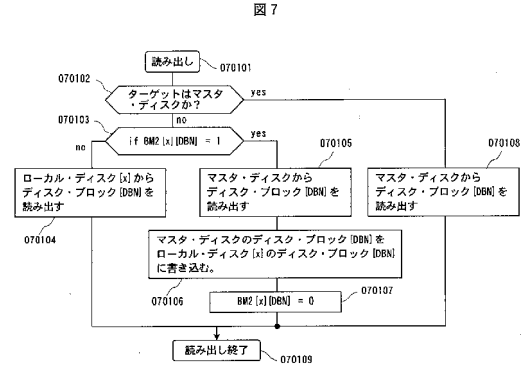
【図 4】



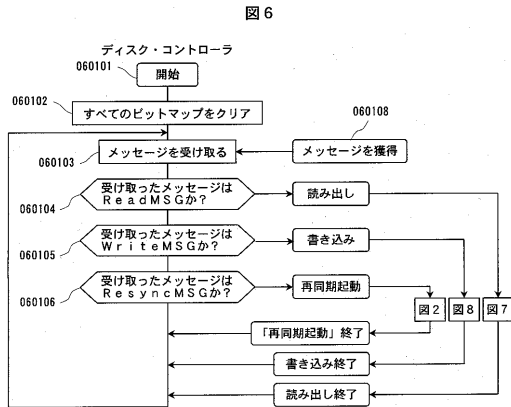
【図 5】



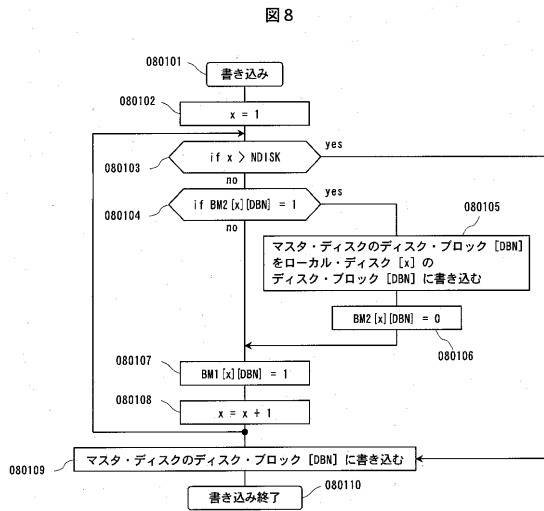
【図 7】



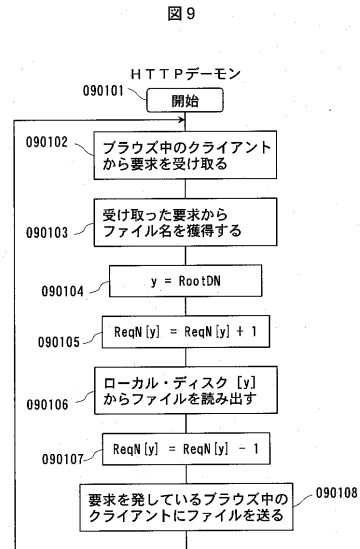
【図 6】



【図 8】



【図 9】



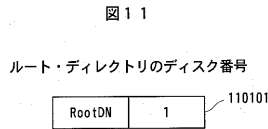
【図 10】

図 10

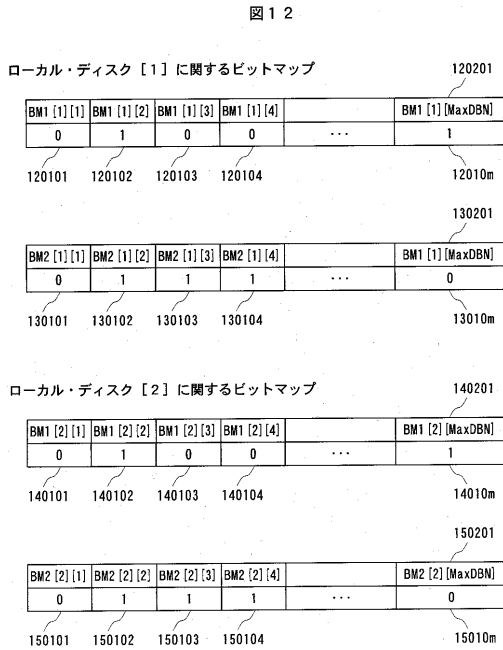
各ローカル・ディスク・ドライブに関する要求の数

ReqN [1]	0	100101
ReqN [1]	3	100102

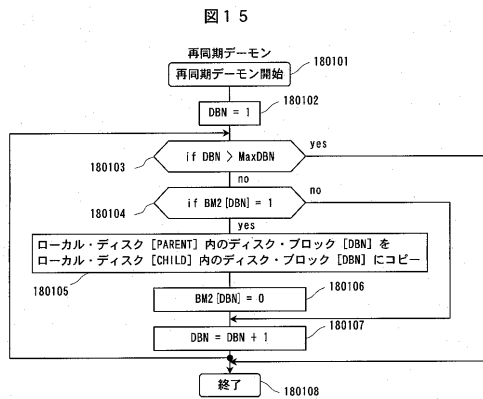
【図 1 1】



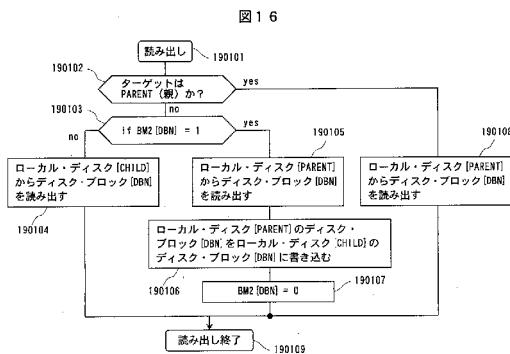
【図 1 2】



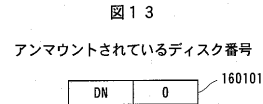
【図 1 5】



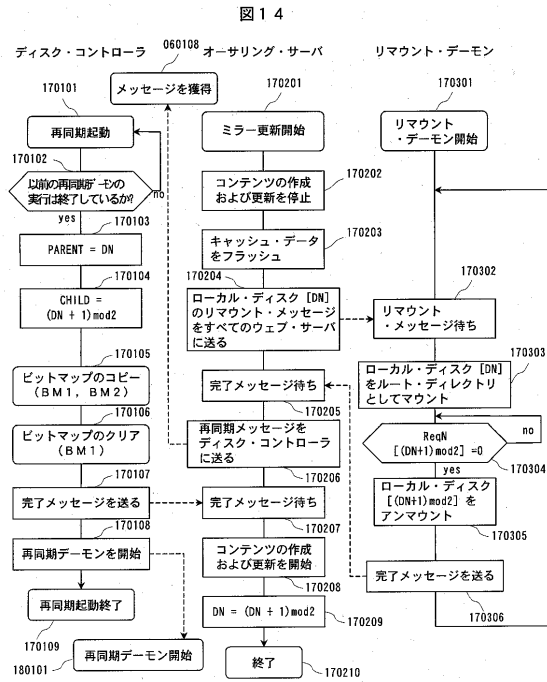
【図 1 6】



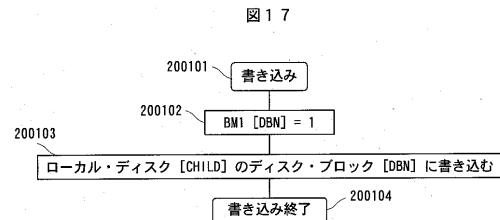
【図 1 3】



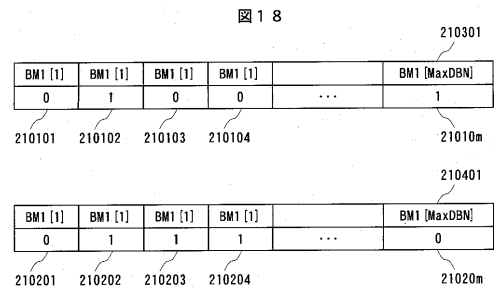
【図 1 4】



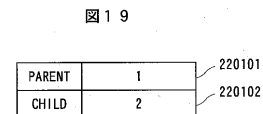
【図 1 7】



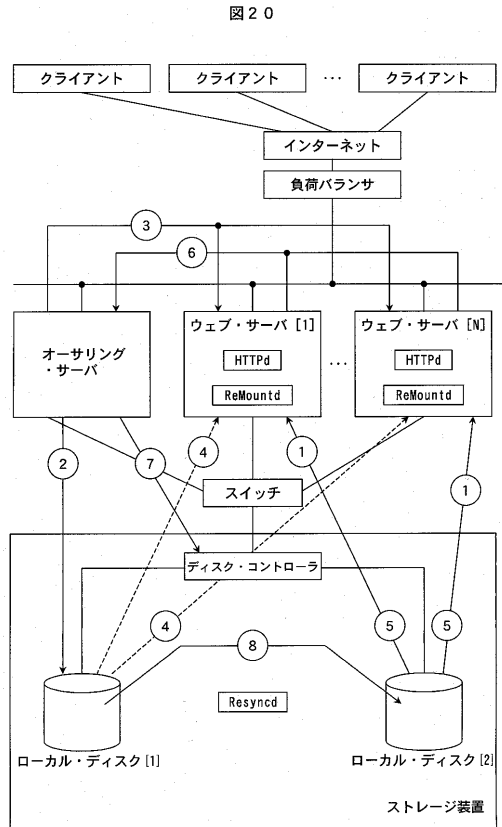
【図 1 8】



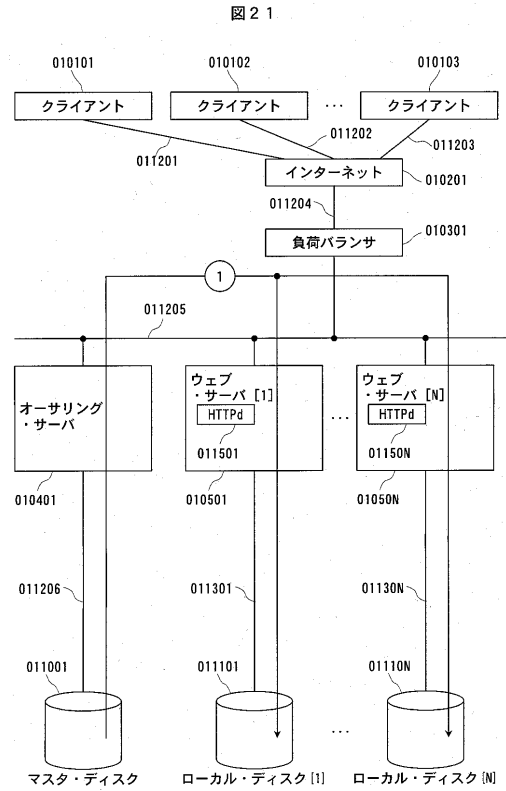
【図 1 9】



【図 20】



【図 21】



フロントページの続き

(72)発明者 山本 政行
アメリカ国 カリフォルニア州サニーベール エルカミノリアル イースト 9 6 5

審査官 浜岸 広明

(56)参考文献 特開平 1 0 - 1 9 8 5 9 0 (J P , A)
特開平 1 0 - 3 3 3 8 3 8 (J P , A)

(58)調査した分野(Int.Cl. , D B 名)
G06F 12/00