(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2004-280779
(P2004-280779A)
(43)公開日　平成16年10月7日(2004.10.7)

(51)Int.Cl.[7]	F I		テーマコード（参考）
G06F 3/06	G06F 3/06	304F	5B065
	G06F 3/06	301M	

審査請求 未請求　請求項の数 26　ＯＬ　外国語出願　（全 46 頁）

(21)出願番号　特願2003-358734(P2003-358734)
(22)出願日　平成15年10月20日(2003.10.20)
(31)優先権主張番号　10/286513
(32)優先日　平成14年10月31日(2002.10.31)
(33)優先権主張国　米国(US)

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(74)代理人　100075096
弁理士　作田　康夫
(72)発明者　渡辺　直企
アメリカ合衆国カリフォルニア州サニーベイル　イーストエルカミノリアル９６５

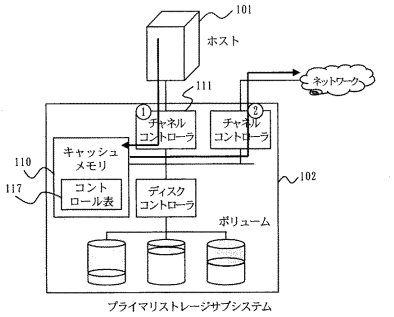
Ｆターム(参考)　5B065 BA01　CE01　EA34

(54)【発明の名称】ストレージシステムのリモートコピーに於いて空白データを省略する方法と装置

(57)【要約】　（修正有）
【課題】ストレージシステムの性能を向上する方法と装置を提供する。
【解決手段】セカンダリサブシステムにイニシャルコピーを作成するとき、あるいはプライマリストレージサブシステムに最初にデータを保存するときに、空白データを省略する。非空白データを順番に送付することにより、欠落しているアドレスが空白データであると識別する、あるいは空白データの存在する位置を指定する省略メッセージにより、データを省略することができる。
【選択図】図５

【図５】

図５



【特許請求の範囲】
【請求項１】
プライマリストレージシステムからセカンダリストレージシステムへ情報をコピーする方法において、
コピーされる情報をプライマリストレージシステムで解析し、情報がコピーされるべきデータで構成されているか、コピーする必要のない空白データで構成されているかを判断すること、
セカンダリストレージシステムへコピーするべきデータを転送し、空白データは転送しないこと、
セカンダリストレージシステムではコピーするべき情報をセカンダリストレージシステムのストレージ装置に書き込みを行うことを有することを特徴とする方法。
【請求項２】
請求項１に記載の方法において、
セカンダリストレージシステムがストレージ装置にコピーを行うステップの前に、ストレージ装置が空白データを有するようフォーマットするステップを有することを特徴とする方法。
【請求項３】
請求項１に記載の方法において、
空白データがプライマリストレージシステムに存在しセカンダリストレージシステムへは転送されないことを示す省略メッセージをセカンダリストレージシステムに送信することをさらに有することを特徴とする方法。
【請求項４】
請求項３に記載の方法において、
省略メッセージを受信後、セカンダリストレージシステムがストレージ装置に空白データを書き込むことを特徴とする方法。
【請求項５】
請求項１に記載の方法において、
転送ステップが、セカンダリストレージシステムへ空白データが転送されたならば前記空白データが保存されたはずのセカンダリストレージシステムの場所に書き込むべき空白データのパターンをセカンダリストレージシステムへ転送するステップを、さらに有していることを特徴とする方法。
【請求項６】
請求項３に記載の方法において、
省略メッセージが、セカンダリストレージシステムへ空白データが転送されたならば前記空白データが保存されたはずのセカンダリストレージシステムの場所に書き込むべき空白データのパターンを、さらに有していることを特徴とする方法。
【請求項７】
請求項１に記載の方法において、
プライマリストレージシステムがディスクコントローラを有し、前記ディスクコントローラがコピーする情報を解析して前記情報がコピーするべきデータから構成されているかあるいは空白データであるかを判断するステップを実行することを特徴とする方法。
【請求項８】
請求項１に記載の方法において、
コピーするべきデータがプライマリストレージからセカンダリストレージシステムへ順番に送られ、そして
セカンダリストレージシステムは前記順番にギャップがあった場合空白データを書き込むことを特徴とする方法。
【請求項９】
請求項１に記載の方法において、情報解析のステップが、前記情報がプライマリストレージシステムのバスを転送されている間に実行されることを特徴とする方法。

【請求項１０】
請求項９に記載の方法において、ＤＭＡコントローラがバスをチェックし情報を解析することを特徴とする方法。
【請求項１１】
ホストと前記ホストに接続するストレージシステムを有し、前記ストレージシステムは前記ホストに接続するチャネルコントローラとデータを保存する装置を有するシステムにおいて、前記装置にデータを書き込む方法であって、
ストレージシステムへデータが転送される際にモニターすること、
データブロックが空白データのみからなる場合、前記装置に省略メッセージを送って前記装置が空白データの書き込みを行うのを抑止すること、及び
データブロックが空白でないデータで構成されている場合、前記データを前記装置に送って前記データの書き込みを行うことを有することを特徴とする方法。
【請求項１２】
請求項１１に記載の方法において
非空白データを書き込む前に前記装置をフォーマットして空白データを保持させるステップが実行されることを特徴とする方法。
【請求項１３】
請求項１１に記載の方法において
省略メッセージが空白データのアドレスを示していることを特徴とする方法。
【請求項１４】
請求項１３に記載の方法において省略メッセージが空白データの量も示していることを特徴とする方法。
【請求項１５】
請求項１3に記載の方法において、省略メッセージを受信後、空白データのアドレスに前記装置が空白データを書き込むことを特徴とする方法。
【請求項１６】
請求項１１に記載の方法において、省略メッセージがさらに空白データのパターンと空白データのアドレスを有していることを特徴とする方法。
【請求項１７】
請求項１６に記載の方法において、省略メッセージを受信後、前記装置が空白データのパターンを空白データのアドレスに書き込むことを特徴とする方法。
【請求項１８】
請求項１１に記載の方法において、ストレージサブシステムはディスクコントローラを有しており、前記ディスクコントローラがコピーする情報を解析し空白データで構成されているか否かを判定するステップを実行することを特徴とする方法。
【請求項１９】
コンピュータの制御下で動作しており、保存する情報と、前記情報を保存する指定されたアドレスを受信するために接続しており、前記情報は空白データと非空白データを有している、ストレージシステムにおいて、受信したアドレスに欠落の有無を判定するため指定されたアドレスをモニターすること、及び、欠落しているアドレスには空白データを書き込むことを有する方法。
【請求項２０】
請求項１９に記載の方法において、指定されたアドレスには非空白データを書き込むことをさらに含むことを特徴とする方法。
【請求項２１】
プライマリストレージシステムにおいて、プライマリストレージシステムからセカンダリストレージシステムへ送るデータを保存されたデータから選択する方法において、
保存されているデータから選択された量のデータを読み取ること、
保存されているデータが空白データか非空白かを判定すること、
保存されているデータが非空白データであれば、前記データをセカンダリストレージシ

ステムへコマンドとともに送付し、セカンダリストレージシステムで前記データを保存させること、
保存されているデータが空白データであれば、前記データはセカンダリストレージシステムへは送付しないこと、及び
保存されているデータが全てセカンダリストレージシステムへ送付されるかあるいは空白データと判定されるまで先行のステップを繰り返すことを有することを特徴とする方法。
【請求項２２】
請求項２１に記載の方法において、
保存されているデータが空白データであった場合に前記データをセカンダリストレージシステムへは送付しないステップが、セカンダリストレージシステムへ空白データが送付されなかったことを表示するメッセージを送ることを含むことを特徴とする方法。
【請求項２３】
請求項２２に記載の方法において、メッセージが空白データのアドレスと量を表示していることを特徴とする方法。
【請求項２４】
プライマリストレージシステムから送信されたデータを受信するために接続しているセカンダリストレージシステムにおいて、プライマリシステムから送信されたデータを保存する方法であって、
プライマリストレージシステムから送信されるデータを保存することを指示するコマンドをプライマリストレージシステムから受信すること、
プライマリストレージシステムから送信されたデータを受信すること、
送付された選択された量の各データの一連のアドレスを受信すること、
一連のアドレスを解析して、アドレスの順番にアドレスの欠落の有無を判定すること、
アドレスで指定された場所にデータを保存すること、
アドレスの順番にて欠落しているアドレスに空白データを保存すること、及び　送付されたデータが全て保存されるまで以上のステップを繰り返すことを有する方法。
【請求項２５】
請求項２４に記載の方法において、
アドレスの順番にて欠落しているアドレスに空白データを保存するステップが、アドレスに指定された場所にデータを保存するステップより以前にセカンダリストレージシステムをフォーマットすることで実行されることを特徴とする方法。
【請求項２６】
請求項２４に記載の方法において、
アドレスの順番にて欠落しているアドレスに空白データを保存するステップが、選択されたデータパターンをアドレスの順番にて欠落しているアドレスに書き込むことで実行されることを特徴とする方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明はストレージシステムに関し特にリモートコピー処理に関する。
【背景技術】
【０００２】
[01] 最新のストレージシステムでは、膨大な量のデータをネットワーク上の多様なローカル及びリモートサイトに迅速に保存し、かつバックアップする機能をユーザーに提供する。このようなシステムでは、プライマリストレージサイトのディスクのイニシャルコピーはセカンダリストレージサイトのディスクに作成される。通常、ストレージシステムでは、ディスク全体のコピーを自動的に取るように設定されており、プライマリサイトとリモートサイトのディスクがリモートコピーのペアに設定されている。この処理を実行するにあたり、通常プライマリサイトのディスクのイニシャルコピーがセカンダリサイトの

対応するディスクに作成される。プライマリサイトでの処理でデータが更新されると、更新されたデータがプライマリサイトからセカンダリサイトに転送されて、セカンダリサイトに“ミラー”される。

【０００３】

[02] しかしながら、データ処理システムの顧客あるいはユーザーの多くは、設置しているハードディスクドライブの容量いっぱいを使用してはいない、ということは良く知られたことである。したがって、各ハードディスクドライブには、セカンダリドライブにバックアップあるいは転送するべきデータのみならず、大量の“空白”データが存在する。この空白データは、ハードディスクのその時点で顧客が使用していない部分の全てを意味する。例えば、空白データは、ディスクのフォーマットされたが使われていない部分から構成されている場合もある。

【発明の開示】

【発明が解決しようとする課題】

【０００４】

[03] ハードディスクの容量が増大し続けており、プライマリサイトのハードディスク全体をセカンダリサイトのハードディスクへリモートコピー処理でコピーする場合、空白データの転送と保存にかなりの時間とネットワーク資源が消費される。空白データの大量コピーは、システムの資源とネットワークの処理能力を使い果たし、一般的に望ましくないことである。

【０００５】

[04] データ圧縮技術により問題は若干緩和される。データ圧縮技術によりネットワーク上を送信されるデータを圧縮してネットワークのトラフィックを軽減し、処理能力の消費が軽減される。このストレージシステムの圧縮技術の例として、米国ミネソタ州のミネアポリスにあるCNTが販売しているUltraNet Edge Storage Router™、あるいはニュージャージ州ランバートンにあるInrange Technologies Corpが販売する9801SNS Storage Network System™があげられる。これらの製品により、送信されるデータ量が減る意味で、ネットワークの負荷は減少するが、送信及び受信する装置は圧縮及び復元する負荷が増大し、復元されたデータには依然として空白データが含まれており、ターゲットのストレージサブシステムに書きこまなければならない。

【０００６】

[05] 空白データの送信とコピーを排除して、ネットワークと送信及び受信システムの負荷を減らす手段と装置が求められている。

【課題を解決するための手段】

【０００７】

[06] 本発明は、ストレージネットワークにおいて、空白データをコピーする必要を排除する方法と装置を提供するものである。本発明により、コントローラのリモートコピー処理の負荷が減少し、空白データの送信の必要が排除され、ネットワークの負荷が減少する。本発明は多様な実現方法を有している。１つの実現方法では、プライマリストレージサブシステムのディスクコントローラが、ディスクからキャッシュメモリにデータを移動するときに空白データを検出する。コントローラはチャネルコントローラに空白データを報告し、空白データがネットワークを通って送られることはなくなる。他の実現方法では、ディスクコントローラがホ、ストチャネルからキャッシュメモリにデータを移動するときに空白データを検出し、空白データの存在をチャネルコントローラに報告し、これにより空白データが送信されることを排除する。

【０００８】

[07] 空白データが検出されれば、リモートコピーコマンドを処理する方法は多種存在する。１つの方法は、空白データを単純に受信側のストレージシステムに送信しないことで、イニシャルリモートコピーでは十分である。この方法では、リモートコピー処理が始まる前に、セカンダリストレージサブシステムのリモートペアを構成するディスクがフォーマットされる。次に、“良い”情報がセカンダリサブシステムのミラーディスクにネッ

トワークを通して送られる。セカンダリシステムのディスクはフォーマットされているので、空白でないデータの記録された個所はすべて有効で、他の個所すべてには空白データが入る。リモートディスクへは、例えばオールゼロでもオール１でも他のどのような形式でも、当該ディスクのフォーマットされた形式にしたがって、空白データが記録される。

【０００９】

[08] 空白データを処理するもう１つの方法は、イニシャルリモートコピー処理時に、全ての良いデータのアドレスを順番通りに保持することである。良いデータをセカンダリストレージ装置に書きこむ際に、ストレージ装置が良いデータの間の省略された領域に自分で空白データの書きこみを行う。

【００１０】

[09] 第三の方法は、ホストの書き込みにもイニシャルコピーにも有効で、“省略”メッセージを発行する空白データの検出器を有している。プライマリストレージが空白データを検出すると、省略する空白データのアドレスと大きさとパターンを含む省略メッセージを発行する。ターゲットストレージがこれを受信すると、選択された領域へのデータの書き込みが停止されるか、あるいは所期の大きさとパターンのデータが書き込まれる。通常はディスクがフォーマット済であればデータは停止され、でなければ省略メッセージの指示があれば所期の空白データパターンが書き込まれる。

【００１１】

[10] 実施例では、プライマリストレージシステムからセカンダリストレージシステムへ情報をコピーする方法に、コピーする情報をプライマリストレージシステムにおいて分析して、コピーするべき情報で構成されているか、あるいはコピーする必要のない空白データで構成されているか判断し、セカンダリストレージシステムへ空白データは送信せず、コピーするべきデータのみを送信し、セカンダリストレージシステムではコピーするべき情報をセカンダリストレージシステムのストレージ装置に書き込む手順が含まれている。ある実施例では、データを書き込むステップを実行する前に、セカンダリストレージがフォーマットされる。他の実施例では、プライマリストレージ装置で空白データが検出されると、空白データのブロックがプライマリストレージシステムに存在し、セカンダリストレージシステムへは送信されないことを示す、省略メッセージがセカンダリストレージ装置へ送られる。他の実施例では、セカンダリストレージシステムはフォーマットされておらず、コピーするべき情報のみが送信され、セカンダリシステムのストレージコントローラが媒体上の他の位置全部に空白データの書き込みを行う。

【発明の効果】

【００１２】

【発明を実施するための最良の形態】

【００１３】

[28] 図１は代表的な従来技術のストレージシステムであり、プライマリストレージサブシステム102がセカンダリストレージサブシステム103とネットワーク105経由で結合している。一般的にネットワーク105は従来技術のネットワークで、例えばATM, sonet、dark fibreあるいはインタネットプロトコルの通信である。さらに、図１は1台のプライマリストレージサブシステムと１台のセカンダリストレージサブシステムを示しているが、多数でも少数でも好きなだけの数の互いに離れた場所にあるストレージシステムを使用することができる。従来これらのストレージシステムを使用するのは大企業でデータの共有を進める、あるいはバックアップコピーを用意して信頼性を向上するなどのためである。

【００１４】

[29] プライマリストレージシステムは一般的にストレージサブシステム102とチャネル103で結合するホスト101の制御下で動作する。チャネル103はチャネルコントローラ111経由でストレージサブシステム102とインタフェース結合している。通常、別のチャネルコントローラ112がストレージサブシステム102をチャネル104経由でネットワーク105へインタフェース接続する。

【００１５】

[30] セカンダリストレージシステム103は同様に構成され、ホスト106がセカンダリシステム103をチャネル108とチャネルコントローラ122経由で制御している。別のチャネルコントローラ121はネットワーク105に結合しているチャネル109とセカンダリストレージサブシステム103とのインタフェースとなっている。通常、チャネル103、104、108及び109はfibre channel、 ESCON、 SCSIあるいはGEが使用される。チャネル104と109はネットワーク105に結合しており、ネットワーク105は公衆回線でも専用線でも良い。

【００１６】

[31] ストレージシステム102は説明済のチャネルコントローラに加えてコントロール表117の制御で動作するキャッシュメモリ110を有している。内部バス114によりキャッシュメモリ110はチャネルコントローラ111と112およびディスクコントローラ113と通信することができる。例えばfibre channel、 SCSIあるいは ATA busなどの別の内部バス115を介してディスクコントローラ113はストレージボリューム116と通信する。ストレージサブシステムは一般的に同様に構成されている。このようにして、一般的にセカンダリストレージサブシステム103の構成はプライマリストレージサブシステム102の構成と適合したものとなっている。

【００１７】

[32] 一般的にデータはチャネルコントローラとキャッシュメモリを経由してプライマリストレージサブシステムに入出力される。ストレージボリュームに書き込むデータが着信すると一時的にキャッシュメモリに記録され、次にディスクコントローラが読み出してボリュームに書き込む。同様にボリューム116からネットワーク経由で送信するデータが読み出されると、チャネルコントローラに大ブロックで送る前に一般的にキャッシュメモリに保存される。ストレージサブシステムは物理ディスクアドレスをホストから見える論理ディスクアドレスに変換する。

【００１８】

[33] 上記のように通常の動作ではプライマリストレージサブシステム102のボリューム116のデータをセカンダリストレージサブシステム103のボリューム126へ“ミラー”することが望ましい。同じく上記のように、図１などのシステムが通常有するかなりの記憶容量の故に、ボリューム116にはかなりの量の“空白”データがあり得る。この空白データはハードディスク上のフォーマットされ未書き込みの部分あるいはディレクトリ入力が無効の（消去表示されているなど）ファイルまたはファイルの一部の大きな部分を占めている。ストレージサブシステムの性能を改善し、チャネルコントローラとネットワークの負荷を減少させるには、この大量の空白データをプライマリストレージサブシステムからチャネル104、ネットワーク105を経由して、ディスクボリューム126へデータを書き込むチャネル109へ送信しないことが有利である。以下にこれを実現した方法を説明する。

【００１９】

[34] 図２は、例えば図１の111,112,121あるいは122のどれであっても良いが、チャネルコントローラのブロック構成図である。図２のチャネルコントローラ201は内部にチャネルインタフェース204、中央処理装置206、メモリ205、ダイレクトメモリアクセス回路207及び内部バス203へのインタフェース208を有している。インタフェース204は例えばネットワークあるいはホストのチャネルコントローラに対して高レベルの機能をもったインタフェースとして動作し、一方インタフェース208はチャネルコントローラ201に対してキャッシュメモリあるいはディスクコントローラなどのストレージサブシステムの内部機能へのインタフェースを実行する。ＣＰＵ206はバス209上での通信を介してチャネル201のコンポーネントを制御する。

【００２０】

チャネルインタフェース204はチャネルプロトコルを制御し、チャネルとＣＰＵ206及びＤＭＡ207のデータの送受信を制御し、メモリ205とチャネルインタフェース204との間のデータ転送を制御する。内部インタフェース208は内部バス203上での通信プロトコルを制御し、ＤＭＡコントローラ207とメモリ205の動作に対応して該ライン上のデータを制御する。

【００２１】
[35] 図３はディスクコントローラ301の構成を図示する概略図である。コントローラ301は図１のディスクコントローラ113または123の処理を実行するために使用することができる。ディスクコントローラは２種類のインタフェースを有している。インタフェースの１つ、内部インタフェース304は内部バス302（または図２の203）へのインタフェースである。他のインタフェース308はディスクバス303へのインタフェースで、ディスクバス303に結合しているストレージボリュームとの通信を可能にしている。内部インタフェース304とディスクインタフェース308はバス309を介して接続している。ディスクコントローラはＣＰＵ306とメモリ305及びDMAコントローラ307を有している。これらのコンポーネントは内部バス302とディスクバス303との間の情報の流れを調整している。内部インタフェース304は内部バスプロトコルを制御し、DMAコントローラ307と共に内部バスとメモリ305の間でデータを通信する。
【００２２】
同様にディスクインタフェース308はディスクバス303のプロトコルを制御し、CPU306、DMAコントローラ307の制御の下でメモリ305とディスクバス303の間でデータを通信する。上記のように、実施例ではディスクコントローラ301はストレージボリューム中の空白データをディスクバス303と接続するストレージボリューム中の空白データを検出することができる。DMAコントローラ307は公知の技術をバス309のチェックに適用している。DMAコントローラ307が空白データを検出するとバス309に接続する他の機器に警告の信号をバス309に発信する。これについてはさらに以下説明する。
【００２３】
[36] 図４乃至６で、各プライマリ及びセカンダリストレージサブシステムでのイニシャルコピー処理の概要を図示する。図４は、イニシャルコピーを作成するときのプライマリストレージサブシステム102を図示する。最初の動作はディスクコントローラ113がディスク116ｂからディスクデータを読み取りキャッシュメモリ110に送付することである。内蔵ＣＰＵ（図示せず）の制御の下で動作しているチャネルコントローラ112はトランザクションをモニターし、キャッシュメモリ110に適当な量のデータが保存され、あわせてディスクコントローラ113の関連エントリがコントロール表117に入力されると、転送を開始する。
【００２４】
所期の量のデータがキャッシュメモリ110に蓄積されると、チャネルコントローラ112は“書き込み”コマンド（あるいは以下に述べるある種のタイプのシステムの場合はベンダー固有のコマンド）を発行しイニシャルコピーの処理を開始する。チャネルコントローラ112は、イニシャルコピーデータをキャッシュメモリ110からチャネル104経由でネットワーク105へ送付する。データはネットワーク105に送られるとセカンダリストレージシステム（図示せず）へ送信される。
【００２５】
[37] 図５はリモートコピー処理時のプライマリストレージサブシステム102の動作を図示する。このプロセスはホスト101がシステム102に書き込みコマンドを発行することにより起動する。次にチャネルコントローラ111がコマンドを受信し解析する。このコマンドで、チャネルコントローラ111はホスト101から書き込みデータを受信しキャッシュメモリ110に保存する。コントローラ111はまた本リモートコピーに関する情報をコントロール表117に保存する。キャッシュメモリ110からのデータがリモートコピーペアの一方として最終的にディスクに保存されると、チャネルコントローラ112はリモートコピーコマンドをセカンダリストレージサブシステム（図示せず）に発行する。
【００２６】
[38] 図6はチャネル109経由でネットワークからイニシャルコピーまたはリモートコピーデータが着信した際のセカンダリストレージサブシステム103の動作を示す図である。このとき、チャネルコントローラ121は受信したコマンドを解析し、データをキャッシュメモリ120に保存し、コントロール表127に適切なエントリを作成する。ディスクコントロ

ーラ123は定期的にコントロール表127をチェックする。ディスクコントローラ123が、ディスクコントローラで実行すべき処理を発見すると、ディスクコントローラはキャッシュメモリ120からデータを読み取り、ターゲットディスク、この場合ディスク126ｂへデータを書き込む。

【００２７】

[39] 空白データを送る場合、図４の（２）、図５の（２）あるいは図６の（３）に示すステップでの負荷は、本発明の実施例では軽減されることに注目してください。言い換えると、図４乃至６に関する処理は、通常の（空白でない）データを処理する際の説明と同様に実行される。しかしながらこれらの処理は空白データを処理する際には必要のないものである。

【００２８】

[40] 図７は本発明による手段の実施例のフローチャートである。本フローチャートはプライマリストレージサブシステム102（図１参照）で実行されるイニシャルコピーの手順を示している。イニシャルコピー処理が開始する前にプライマリとセカンダリのシステムは同期されリモートコピーペア、すなわちプライマリストレージサブシステムとセカンダリストレージサブシステムのディスク間の対応、が形成される。本処理はシステムアドミニストレータが実行する。本公知の処理の後、図７のイニシャルコピー処理が実行される。

【００２９】

イニシャルコピー処理の実行の最初のステップはファイルシステム701のクリーニングである。本ステップではプライマリディスクの未使用部分とセカンダリディスクの全ては空白データで占められる。このステップは商用入手可能なソフトウェア、例えばフィンランドのタンペールのJeticoから入手可能なBC Wipe^{TM} を使用して実行される。クリーニングソフトウェアと空白データのコピーを省略することを合わせると、ここに述べるようにリモートコピー処理の負荷をかなり減少させることができる。以下の説明で明らかになるとおり、クリーニング処理は必須ではない。

【００３０】

[41] クリーニングが完了すると、ステップ702でイニシャルコピーの作成が開始し、ターゲットディスクからデータが読み出される。本処理ではディスクコントローラ113はディスクからデータを読み出しキャッシュメモリ110に送付する。キャッシュの読み出しと書き込みでは、転送に際して通常大量のデータを、例えば１MBのデータが処理される。転送自体はDMA308の制御下で実行される。

【００３１】

[42] 図７に示すようにデータが読み出されると、ステップ703でデータをチェックして空白データの存在を判定する。本処理はデータパターンを判定するためバストラフィックを監視しているDMA307により実行される。この処理を通常“snooping”と呼んでいる。DMA307がデータパターンが空白データで構成されていることを検出すると、当該情報がCPU306へもたらされる。データが空白であるか否かの判定はクリーニング処理に使われたソフトウェアまたはイニシャルコピー処理を実行する前にハードディスクをフォーマットするのに使用されたソフトウェアで実行される。

【００３２】

通常、空白データはオールゼロか、オール１あるいはゼロと１の繰り返し、例えば010101で構成されている。このパターンが現れたら、CPU306はコントロール表117を用いて空白データのコピーを省略するようにシステムに指示する。一方、データが１とゼロのランダムなパターンであったときは、CPU306はコントロール表117にエントリを行って通常処理を可能にする。

【００３３】

[43] データが転送されるとチャネルコントローラ112は定期的にコントロール表117をチェックする。チャネルコントローラが表117を用いて“正常な”データのコピー－イニシャルコピー－であると判断したらチャネルコントローラ112はデータ転送を行う。この

転送でデータは最終的にプライマリストレージシステム102からセカンダリストレージシステム103にコピーされる。CPU206はDMA207にコマンドを送りキャッシュメモリ110からメモリ205へデータを移動させる。その後、CPU206はチャネルインタフェース装置112へコマンドを送り、イニシャルコピー（リモートコピー）コマンドをセカンダリストレージシステム103へネットワーク105を通して発行させる。リモートコピーコマンドの発行はステップ704に示されている。

【００３４】

[44] 空白データのコピーを省略要求のときはCPU206は省略メッセージを発行しデータ転送を行わない。省略メッセージ自体が１以上の量（１MBサイズ、上記参照）のデータがコピーされないことを示している。ディスクコントローラはディスク上のすべてのデータがセカンダリサブシステムに送られるか、空白データと判定されるまで、図７のステップ702－705を繰り返し続ける。これはステップ706に示されている。

【００３５】

[45] ステップ707に示されるように、ディスク上の全てのデータが転送されるか空白データと判定されるとディスクコントローラは次のディスクまたはボリュームを選択して処理を繰り返す。最後にステップ708に示すように、プライマリサイトのストレージの全ボリュームがセカンダリサイトのストレージにイニシャルコピーされるかリモートコピーされる。この時点でプライマリストレージサイト102のイニシャルコピー処理は完了する。

【００３６】

[46] 図８はセカンダリストレージシステム103のイニシャルコピー処理を示すフローチャートである。上記のごとく、管理者の最初のステップの選択肢はステップ801に示すようにセカンダリストレージのボリュームをフォーマットすることである。セカンダリストレージサイトでのイニシャルコピーの実行を指示するコマンドがステップ802で受信されると処理が開始される。このとき、チャネルコントローラ121はイニシャル（リモート）コピーコマンドをネットワーク105を介して受信する。CPU206はチャネルインタフェース204で受信したコマンドを解析する。コマンドが通常データのコピーを指示するものであったときはCPU206はチャネルインタフェース204にプライマリストレージシステム102からネットワーク105経由でデータを受信するように指示する。データが着信するとチャネルインタフェース204はデータをメモリ205に保存する。

【００３７】

[47] 次にCPU206の制御の下に、DMA装置207はデータをメモリ205からキャッシュメモリ120へ移動するよう指示を受ける。CPU206からの本要求はコントロール表127に保存される。ディスクコントローラ123は定期的にコントロール表127をチェックする。ディスクコントローラ123が要求を発見すると、ディスクインタフェース308はCPU306の指示を受け、書き込みコマンドを発行しディスクにデータを書き込む。これはフローチャートにステップ803,804及び806として示されている。以上は正常（空白でない）データの処理である。

【００３８】

[48] 一方、チャネルコントローラ121が受信したコマンドが空白データを意味する省略メッセージの場合、CPU206は省略処理を開始する。このようにステップ803で空白データが検出され、ステップ805で空白データの省略処理が実行される。この場合、セカンダリストレージシステム103は事前にディスクがクリーニングまたはフォーマットされており、すでに空白データが指定された位置に書き込まれているので、データをディスクに書き込むことは行わない。省略メッセージによりある種のパターンのデータを書き込むことを求められ、ディスクがクリーニングまたはフォーマットされておらず、そのパターンがセカンダリストレージに書き込まれていない場合はディスクコントローラ123は省略要求を検出すると指定されたパターンをディスクに書き込む。

【００３９】

本処理はCPU306が所期のパターンをメモリ305に生成し、ディスクインタフェース装置308にそのデータをディスクに書き込むようコマンドを発行することにより達成される。上

記のプロセスはステップ807に示すようにセカンダリストレージサイト103のイニシャルコピーが完了するまで繰り返し実行される。

【００４０】

[49] 前に説明したように、セカンダリストレージサイトで読み出しと書き込みを実行中に空白データを省略する手段はいくつか存在する。空白データを省略する方法のうち２つを、図9ｂと9ｃに示している。図9ａは空白データを省略しない場合の手順を示している。図９ａを最初に説明する。図９ａにてＤａｔａ0とＤａｔａ３は正常データで、Ｄａｔａ１とＤａｔａ２は空白データである。

【００４１】

図9ａに示すように、通常のイニシャルまたはリモートコピー処理では、プライマリストレージサブシステム102はデータセットＤａｔａ０、Ｄａｔａ１、Ｄａｔａ２、Ｄａｔａ３及びＤａｔａ４の各々に計４つのコピーコマンドを発行する。これらのコマンドによりデータはプライマリストレージシステム102のメモリからセカンダリストレージシステム103のメモリへ転送される。

【００４２】

セカンダリストレージシステムではさらに４つのコマンドが解読され実行され、Ｄａｔａ０、Ｄａｔａ１、Ｄａｔａ２及びＤａｔａ３がセカンダリストレージシステムのディスクに書き込まれる。上記の説明から明らかなように、Ｄａｔａ１及びＤａｔａ２をプライマリストレージシステムの中からネットワークを介して送信しセカンダリストレージシステムに再度書き込みを行うために使用される時間と通信資源は不必要であり、無駄である。

【００４３】

[50] 図9ｂと9ｃに空白データを省略する２つの技術を図示している。図9ｂはセカンダリストレージサブシステムのボリュームが、プライマリストレージシステムの空白データのフォーマットと一致するようにフォーマットされている場合の空白データを省略する処理を示す図である。図9ｂの場合、上記図7と8を用いて説明した技術により、プライマリストレージシステムがＤａｔａ１及びＤａｔａ２が空白データであり、これらをセカンダリストレージシステムに送信する必要がないことを検出する。

【００４４】

セカンダリストレージシステムは既に適切にフォーマットされているのが分かっているので、プライマリストレージシステムはＤａｔａ１またはＤａｔａ２を送信することも、あるいはＤａｔａ１またはＤａｔａ２の存在を通知する必要もない。Ｄａｔａ0及びＤａｔａ3がセカンダリストレージサブシステムボリュームにコピーされるとこれらのボリュームの内容はプライマリストレージシステム102の対応するボリュームの内容と一致（ミラー）する。

【００４５】

[51] セカンダリストレージシステム103がフォーマットされていない場合、他の代替手段が実行される。この場合、プライマリストレージシステムがイニシャルコピーコマンドを順番に発行すると、セカンダリストレージシステムはＤａｔａ1とＤａｔａ2のアドレスが省略されたことが分かる。この場合、空白のＤａｔａ1とＤａｔａ2はネットワークを介して送信されることはないが、セカンダリストレージシステムはＤａｔａ1とＤａｔａ2の省略されたアドレスを検出できるので、ストレージボリュームに適切な形式で空白データの書き込みを行う。

【００４６】

[52] 図9ｃは本発明の他の実施例を示す。図9ｃでは空白データを省略したことが、プライマリストレージサブシステム102からセカンダリストレージシステム103へ空白データが省略されたことを示すメッセージの送信により連絡される。この場合、セカンダリストレージサブシステムのボリュームを前もってフォーマットしておく必要はない。図に示すようにＤａｔａ1とＤａｔａ2から構成される空白データの存在はプライマリストレージサブシステムにより、例えばＤＭＡ装置がデータの得られるバスをチェックしているなどに

より検出される。データのあるものが、例えばＤａｔａ1とＤａｔａ2が、空白データであると判定されると、プライマリサブシステム102は１つのメッセージにそれらをまとめてセカンダリストレージサブシステム103へその情報を送る。セカンダリストレージサブシステムはプライマリストレージサブシステム102から送られてきた省略メッセージを解析して、ストレージボリュームの中の省略メッセージで指定されたＤａｔａ1とＤａｔａ2の位置に空白データの書き込みを実行する。

【００４７】

省略メッセージの送信によりセカンダリストレージシステムでの該当するデータの書き込みの必要性はなくならないが、ネットワークのトラフィックは軽減し、セカンダリサイトでの負荷が軽減し、“真の”データをプライマリとセカンダリのサブシステム間で送信することに資源をあてることが可能になる。

【００４８】

[53] 図９ｄは他の観点から空白データの省略処理を説明する図である。図の上側はデータ（空白および非空白）及びデータをセカンダリシステムに書き込みを行う通常コマンドが示されている。コマンド(Cmd)は指定されたサイズとアドレス(Adr)のデータブロックの書き込み指示する。例えば、最初のコマンドはサイズ0ｘ0010でアドレス0ｘ0030にデータの書き込みを行うものである。図の下側の部分は非空白データのみをセカンダリサブシステムに送る方法を示している。

【００４９】

[54] 図１０ａは省略メッセージを示す図である。図１０ａの左側の、プライマリとセカンダリストレージシステム間で空白データを転送するための、データ転送の数に注目して欲しい。図１０ａの右側は、指定されたデータの“塊”が省略されたことを示すために省略メッセージが使われている。データ(FCP_DATA)を送信するステップが全て削除されている。

【００５０】

[55] 図１０ｂは図１０ａと同様の図である。図１０ｂの左側に示すように、空白データが送信されると膨大な量の空白データがプライマリからセカンダリストレージシステムへ送信される。逆に図１０ｂの右側では本発明の方法が適用され両ストレージサブシステム間で全空白データの送信の必要がなくなっている。空白データの転送は図１０ａで説明したのと同じコマンドで示されている。

【００５１】

[56] 図11は省略メッセージの１例である。メッセージは９バイト（行で表す）で構成され、９バイトの各ビットの状態は列で表されている。オペレーションコードはコピーを省略することを示しており、バイト２－５に開始論理ブロックアドレスがバイト７－８に転送長が示されている。

【００５２】

[57] 図１２は省略メッセージの例で、セカンダリストレージサブシステムサイトで書き込むべきデータパターンが含まれている。省略メッセージは図11と同じフォーマットであるがストレージボリュームに書き込むべきデータパターンがメッセージのバイト6に表示されている。図11と図12に示すメッセージフォーマットは代表例である。メーカー固有のメッセージを使用することができ、特定のメッセージあるいはコマンドに基づいて、多様なサイズの空白データを省略することができる。例えば、次の１０バイト、次の１００バイトのコピーを省略するなど個々のメッセージを送ることができる。

[58] 図13はチャネルコントローラを使用してプライマリストレージサブシステムにホストが書き込みを行う場合の手順を示すフローチャートである。この場合、書き込み処理で空白データが検出されるとプライマリシステムでの最初のデータ保存を加速することができる。チャネルコントローラ111のＣＰＵ206がホストの入出力動作を処理し、これはステップ1301及び1303に示されている。チャネルコントローラ112のＣＰＵ206はリモートコピーを処理し、これはステップ1303,1304及び1305に示されている。

【００５３】

[59] ホスト101が“書き込み”コマンドを発行すると、チャネルインタフェース204がコマンドを受信しＣＰＵ206に報告し、場合によりメモリ205にコマンドを保存する。ＣＰＵ206はコマンドを解析し有効であれば実行を開始する。無効であればＣＰＵ206は“no good”ステータスをチャネルインタフェース204経由でホスト101に返す。これらの動作は例えばSCSI仕様に従って処理される。

【００５４】

[60] コマンドが有効であると仮定して、ＣＰＵ206はホスト101からのデータ転送の開始をインタフェース装置204に指示する。インタフェース装置204はチャネル103からメモリ205へデータを転送する。ＣＰＵ206はメモリ205へ指定された量のデータが転送されるのを待ち、次いでＤＭＡ207にコマンドを発行する。ＤＭＡ207は内部インタフェース208と内部バス203を使用してメモリ205からキャッシュメモリ110へデータを転送するよう指示を受ける。ＣＰＵ206はインタフェース208を制御してチャネルインタフェース204を追い越すことを防止している。

【００５５】

すべてのデータがキャッシュメモリ110に転送されるとＣＰＵ206はチャネルインタフェース204とチャネル103を経由してホスト101に”good”ステータス報告を返す。これらの処理の実行にともないＣＰＵ206は定期的にコントロール表117を更新する。これらの動作すべては図13の書き込みコマンドステップ1301に含まれている。

【００５６】

[61] 上記のように転送が行われている間ＤＭＡ207はデータパターンをチェックしている。データが空白データであった場合、ＤＭＡ207はＣＰＵ206に省略メッセージをコントロール表117に書きこませる。ディスクコントローラ113のＣＰＵ306はデータ書き込み中に定期的にコントロール表117をチェックしている。ＣＰＵ306が、ディスクコントローラ113で処理される省略メッセージを検出すると、ＣＰＵ306はデータの書き込み処理を開始する。処理のために必要な情報はここに全てコントロール表117に含まれている。

【００５７】

ＤＭＡ307はＣＰＵ306の指示を受け、チャネルコントローラ111がコントロール表117をチェックする速度を追い越さない速度でデータをキャッシュメモリ110からメモリ305に転送する。データ転送の間ＤＭＡ307はまた空白データを検出することができる。ＣＰＵ306は指定された量のデータがメモリ305に転送されるまで待つ。データが空白データであった場合、ＣＰＵ306は省略メッセージをコントロール表117に入力する。ＣＰＵ306は次にディスクインタフェース装置308とディスクバス303経由でメモリ305からディスク115にデータの転送を開始する。

【００５８】

ＣＰＵ306はディスクインタフェース308を制御し、内部インタフェース304を追い越すことがないことを保証している。全てのデータがディスク115に書き込まれると、これもステップ1302に示すように、ＣＰＵ306はリモートコピー情報をコントロール表117に発行する。

【００５９】

[62] チャネルコントローラ112は定期的にコントロール表117をチェックしている。チャネルコントローラ112がリモートコピー要求を検出するとリモートコピー処理を開始する。2件の連続したリモートコピー処理であれば、ＣＰＵ206は１つのリモートコピーコマンドに結合することができ、これは標準コマンドでもベンダー固有コマンドであっても良い。リモートコピーコマンドを発行する前にＣＰＵ206はコントロール表117をチェックする。省略メッセージがあれば、ＣＰＵ206はステップ1305に示すように省略メッセージを発行する。ＣＰＵ206が省略メッセージを発見しなかった場合、リモートコピーコマンドを発行しセカンダリストレージサブシステム103にデータを転送する。セカンダリストレージサブシステム103でのリモートコピー処理は図8のステップ802－806で説明したのと同様である。

【００６０】

[63] 図14は上記の方法と装置とは異なる処理の概要を示す図である。図14の第一列は個別のアプリケーションを、言い換えると空白データはセカンダリサイトでイニシャルコピーを作成中か、もっと前にプライマリストレージサイトで書き込み中に検出されるのかを示している。図14の次の列はプライマリストレージサイトで空白データを検出するコンポーネントを示している。プライマリストレージサブシステムからセカンダリストレージサブシステムへのイニシャルコピーの場合、空白データはディスクコントローラにより検出される。一方、プライマリストレージシステムにホストから初めてデータが書き込まれるときはチャネルコントローラが空白データを検出する。

【００６１】

[64] 図表の次の列に空白データの検出に対しての対処が要約されている。ディスクコントローラで検出された場合、すなわちイニシャルコピー処理の場合、省略メッセージが生成されるか、データが無視されるかである。これら２つの処理は図9ｂと9ｃを用いてそれぞれ説明されている。ホストから空白データが着信したことをチャネルコントローラが検出した場合、データの書き込みを省略するべきことを下流のディスクに伝えるため省略メッセージが使われる。本処理については図9ｃを用いて説明した。

【００６２】

[65] 図14の右側の列はターゲットサブシステムで生ずる事項を述べている。イニシャルコピー時の場合は、ターゲットサブシステムの対応は全てセカンダリシステムで発生する。省略メッセージの場合、ディスクがすでに適切にフォーマットしてあれば、セカンダリサブシステムでは対応する必要がない。前にフォーマットしてなければ、右側のブロックに示すようにデータの書き込みを行う必要がない。同様にイニシャルコピー処理で、空白データは単に省略する場合、ディスクが以前にフォーマットされていればセカンダリサブシステムでは対応する必要がない。

【００６３】

ディスクが以前にフォーマットされていないかクリーニングされていなければ空白データを書き込まなければならない。図14の最下段の行にホストの書き込み処理について示す。この場合は省略メッセージによりプライマリサブシステムに空白データが書き込まれる結果になる。

【００６４】

[66] 以上は発明の実施例の説明である。以下の請求項の範囲を逸脱することなく発明の実施において変更を加えることが可能であることは容易に分かることである。

【図面の簡単な説明】

【００６５】

【図１】図１は、ネットワーク経由で通信するプライマリストレージサブシステムとセカンダリストレージサブシステムを有するストレージシステムを示す図である。

【図２】図２は、図１にブロック形式で示すチャネルコントローラの構成を示す図である。

【図３】図３は、同じく図１のブロックに示すディスクコントローラの構成を示す図である。

【図４】図４は、プライマリストレージシステムのイニシャルコピー処理を示す図である。

【図５】図５は、この処理を詳細に示す図である。

【図６】図６は、プライマリストレージシステムからのコマンドに応じたセカンダリストレージシステムの動作を示す図である。

【図７】図７は、プライマリストレージシステムのイニシャルコピー処理の手順を示すフローチャートである。

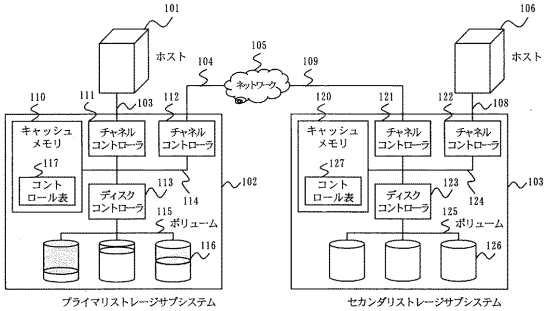
【図８】図８は、セカンダリストレージサブシステムのイニシャルコピーの手順を図示する図である。

【図９ａ】図９ａは、空白データを省略することなくコピーする従来技術の処理を図示する図である。

【図９ｂ】図９ｂは、空白データを省略するリモートコピー処理を図示しする図である。
【図９ｃ】図９ｃは、空白データを省略するメッセージを使用して空白データを省略したリモートコピー処理を図示しする図である。
【図９ｄ】図９ｄは、空白データの送信とコピーの省略を行う処理を図示する図である。
【図１０ａ】図１０ａは、本発明の1実施例におけるプライマリストレージシステムとセカンダリストレージシステムの関係を図示する図である。
【図１０ｂ】図１０ｂは、本発明の1実施例におけるプライマリストレージシステムとセカンダリストレージシステムの関係を図示しする図である。
【図１１】図１１は、ベンダー固有のコマンドを使用した“省略”メッセージの例を示す図である。
【図１２】図１２は、省略メッセージの他の例であるがデータパターンを図示しする図である。
【図１３】図１３は、プライマリストレージシステムとダイレクトメモリアクセス装置の空白データを検出する動作のフローチャートである。
【図１４】図１４は、ストレージシステムの動作の多様な様式を示す図である。
【符号の説明】
【００６６】
101・・・ホスト、105・・・ネットワーク、106・・・ホスト、110…キャッシュメモリ、111・112・・・チャネルコントローラ、113・・・ディスクコントローラ、115・・・ボリューム、120・・・キャッシュメモリ、121・122・・・チャネルコントローラ、123・・・ディスクコントローラ、125・・・ボリューム、204・・・チャネルインタフェース、205・・・　メモリ、206・・・CPU、207・・・DMA、208・・・内部インタフェース、203・・・　内部バス、304・・・内部インタフェース、305・・・メモリ、306・・・CPU、307・・・DMA、308・・・ディスクインタフェース、303・・・ディスクバス

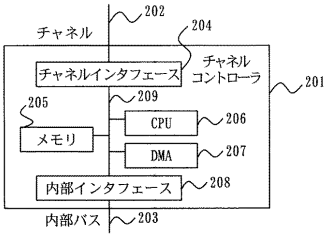
【図１】
【図1】

図１　（従来技術）



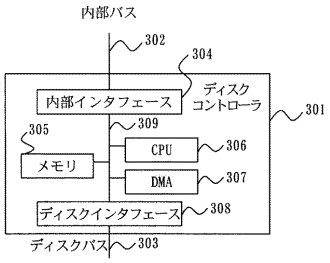
【図２】
【図2】

図２



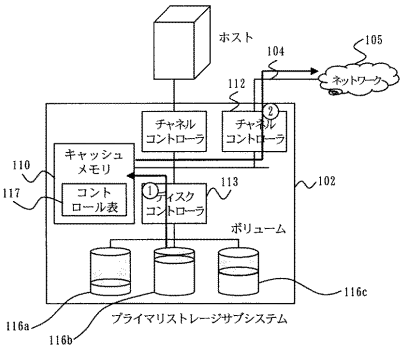
【図３】
【図3】

図３



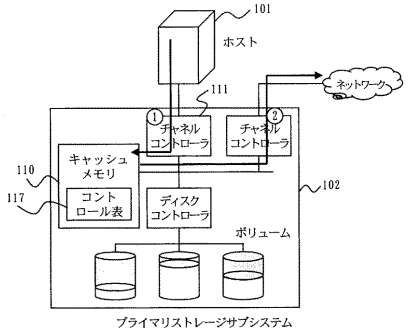
【図４】
【図4】

図４



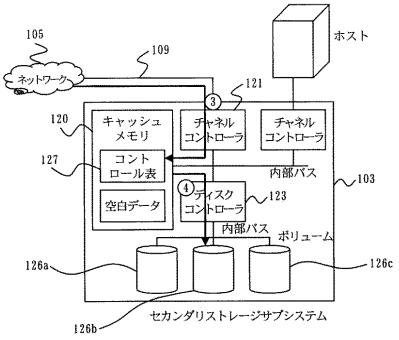
【図５】
【図5】

図5



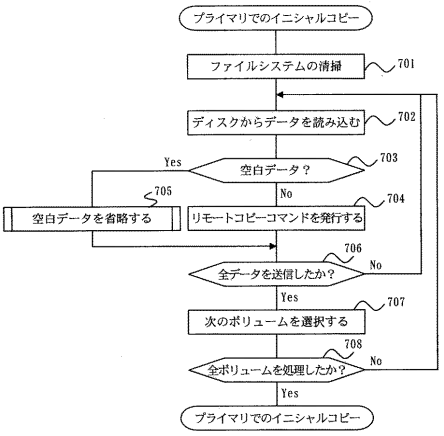
【図６】
【図6】

図6



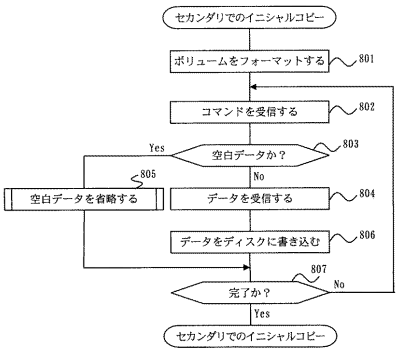
【図７】
【図7】

図7

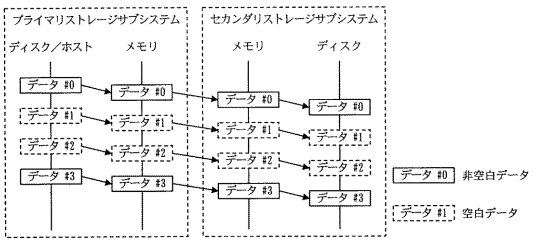


【図８】
【図8】

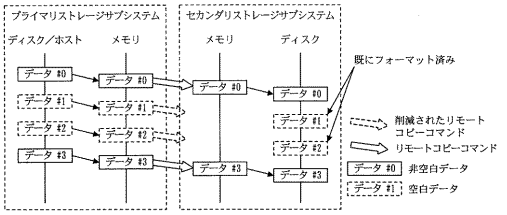
図8



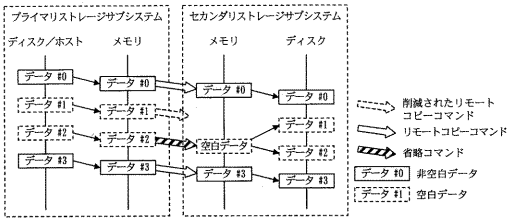
【図９ａ】
【図9ａ】
図９（ａ）　リモートコピー－空白データを省略しないとき（従来技術）



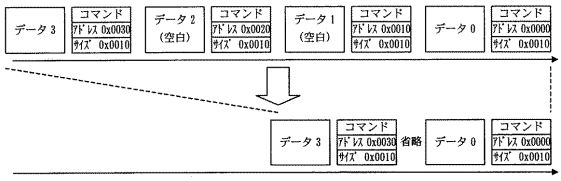
【図９ｂ】
【図９ｂ】
図９（ｂ）　空白データを省略したリモートコピー



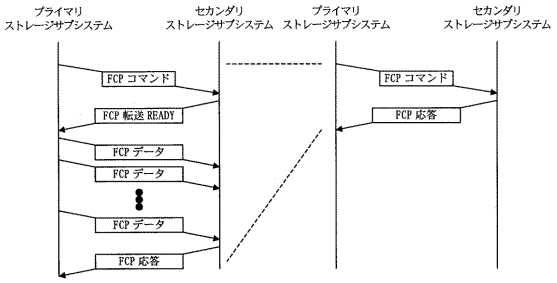
【図９ｃ】
【図９ｃ】
図９（ｃ）　省略メッセージにより空白データを省略したリモートコピー



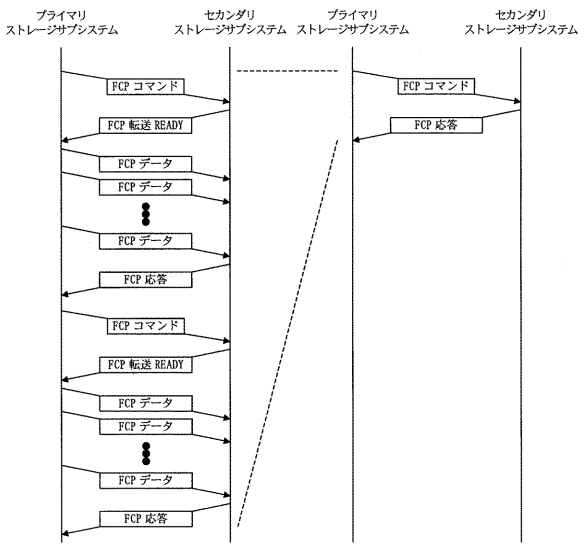
【図９ｄ】
【図９ｄ】
図９（ｄ）　空白データの省略を示すリモートコピー



【図１０ａ】
【図１０ａ】
図１０（ａ）　空白データを省略するプロセス



【図１０ｂ】
【図１０ｂ】
図１０（ｂ）　継続する空白データを省略するプロセス



【図１１】
【図１１】
図１１

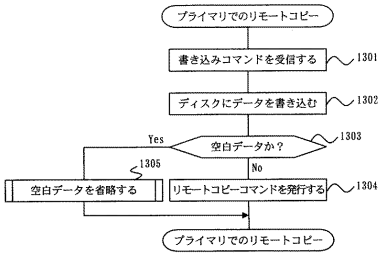
ビット バイト	7	6	5	4	3	2	1	0
0	動作コード（0xCA）							
1	予　備			DPO	FUA	予　備		相対ｱﾄﾞﾚｽ
2	(MSB)							
3				論理ブロックアドレス				
4								
5								(LSB)
6	予　　備							
7	(MSB)							
8	転送サイズ							(LSB)
9	管　理　用							

【図１２】
【図１２】
図１２

ビット バイト	7	6	5	4	3	2	1	0
0	動作コード（0xCA）							
1	予　備			DPO	FUA	予　備		相対ｱﾄﾞﾚｽ
2	(MSB)							
3				論理ブロックアドレス				
4								
5								(LSB)
6	パ　タ　ー　ン							
7	(MSB)							
8	転送サイズ							(LSB)
9	管　理　用							

【図１３】
【図１３】

図１３



【図１４】
【図１４】

図１４

#	アプリケーション	空白データをプライマリで検出する	空白データ情報	ターゲットサブシステムでの対応
1	イニシャルコピー	ディスクコントローラ	省略メッセージ	空白データを書き込む
2				何もしない（フォーマット済み）
3			単に省略する	空白データを書き込む
4				何もしない（フォーマット済み）
5	ホスト書き込み	チャネルコントローラ	省略メッセージ	プライマリで空白データを書き込む

【外国語明細書】

PATENT

Attorney Docket No. 16869B-049500
Client Ref. No. HAL-ID 226

Method and Apparatus for Skipping Remote Copy of Null Data in Storage Systems

BACKGROUND OF THE INVENTION

[01] Modern storage systems provide users with the capability of storing and backing up enormous amounts of data quickly over networks among various local and remote locations. In such systems, an initial copy of disks at a primary storage site is made to disks at a secondary storage site. The storage system is typically configured to automatically copy the entire disk, and configure the disks at the primary and remote storage sites as remote copy pairs. In performing these operations, typically an initial copy of the disk at the primary site is made to a corresponding disk at the secondary site. As operations at the primary site change the data there, that changed data is "mirrored" at the secondary site by transmission from the primary site to the secondary site of the changed data.

[02] It is well know, however, that most customers or users of data processing systems do not use the entire storage capability of the hard disk drives situated at their facility. Thus, each hard disk drive will include information which is desired to be backed up or transferred to the secondary or remote site, as well as large amounts of "null" data. This null data represents the contents of all of the portions of the hard disk which are not being used by the customer at that time. For example, the null data may consist of formatted, but unused portions of the disk.

[03] With the sizes of hard disks continuously increasing, substantial time and network resources are being employed to transfer and store this null data at remote sites when the entire hard disk at a primary site is copied in a remote copy operation to a hard disk at the secondary site. This extensive copying of null data depletes system resources and network bandwidth, and is generally undesirable.

[04] The effect has been alleviated to some extent by data compression techniques. Such data compression techniques can reduce network traffic by compressing the data to be transmitted over the network so that it consumes less bandwidth. Examples of such compression technology in storage systems include the UltraNet Edge Storage Router™ sold by CNT of Minneapolis, Minnesota, or the 9801SNS Storage Network System™ sold by

Inrange Technologies Corp. of Lumberton, New Jersey. Although those products reduce the load on the network in the sense the fewer bits are transferred, they increase the load on the transmitting and receiving units which must compress and decompress the data, and the decompressed data still includes the null data which is written to the target storage subsystem.

[05] What is needed is a method and apparatus which can avoid transmitting and copying null data, thereby reducing the load on the network, the transmitting system and the receiving system.

BRIEF SUMMARY OF THE INVENTION

[06] This invention provides a method and apparatus which eliminates the need to copy null data across a storage network. The invention reduces the load on controllers for the remote copy operation, and reduces the load on the network by eliminating the need to transmit the null data. The invention has a variety of implementations. In one implementation, at the primary storage subsystem, the disk controller detects the null data as it moves data from a disk to a cache memory for transmission. It reports the null data to the channel controller so that the null data is not sent across the network. In another implementation, the disk controller detects the null data as it moves data from the host channel to the cache memory and reports the presence of the null data to the channel controller, thereby again precluding its being transmitted.

[07] Once the null data is detected, there are a variety of approaches for handling the remote copy command. In one approach, satisfactory for the initial remote copy operation, the null data is simply not transmitted to the receiving storage system. In this implementation, before the remote copy operation is begun, the disks at the secondary storage subsystem which are to be configured as remote copy pairs, are formatted. Then, the "good" information is transmitted over the network to the mirroring disk in the secondary subsystem. Because the disks in the secondary system have been formatted, everywhere that the non-null information is stored will be useful, and all other portions of the disk will contain null data. The null data will be stored on the remote disk in whatever format that disk is formatted, e.g., all zeros, all ones, or some other desired pattern.

[08] Another approach to handling the null data is that during the initial remote copy operation the addresses of the all the good data are maintained in order. As the good

data is written to the secondary storage unit, the storage unit itself writes null data to any skipped areas between the good areas.

[09] A third approach, useful for both the host write and the initial copy, has the detector of the null data generating a "skip message." Whenever the primary storage detects null data, it creates a skip message which includes the skipped address, the size and the pattern of the null data. When this is received by the targeted storage, writing of the data into the selected region is blocked, or the desired size and pattern are written. Typically the data will be blocked if the disk has already been formatted, or a desired null data pattern will be written wherever the skip message directs.

[10] In a preferred embodiment, a method for copying information from a primary storage system to a secondary storage system, includes the steps of at the primary storage system, analyzing the information to be copied to determine if it constitutes data desired to be copied or null data which does not need to be copied; transmitting to the secondary storage system the data desired to be copied and not transmitting the null data; and at the secondary storage system, writing to a storage unit in the secondary storage system the information to be copied. In some embodiments the secondary storage unit is formatted before the data is copied to the before the step of copying is performed. In other embodiments when null data is detected at the primary storage unit, a skip message is sent to the secondary storage unit which indicates that a block of null data is present on the primary storage system and will not be transmitted to the secondary storage system. In other embodiments where the secondary storage system has not been formatted, only the information to be copied will be transmitted, but the storage controller at the secondary system will write null data in all other locations on the media.

BRIEF DESCRIPTION OF THE DRAWINGS

[11] Figure 1 is a diagram illustrating a storage system which includes a primary storage subsystem and a secondary storage subsystem which communicate with each other via a network;

[12] Figure 2 is a diagram which illustrates the configuration of a channel controller as shown in block form in Figure 1;

[13] Figure 3 is a diagram illustrating the configuration of a disk controller, also shown in block form in Figure 1;

[14] Figure 4 is a diagram of the primary storage system illustrating its operations in an initial copy process;

[15] Figure 5 is a diagram further illustrating this process;

[16] Figure 6 is a diagram illustrating actions at the secondary storage subsystem in response to commands from the primary storage subsystem;

[17] Figure 7 is a flowchart illustrating steps in an initial copy operation at the primary storage system;

[18] Figure 8 is a flowchart illustrating the process of an initial copy of a secondary storage subsystem;

[19] Figure 9a illustrates the conventional prior art operation in which null data is copied without being skipped;

[20] Figure 9b illustrates the operation of a remote copy where null data is skipped;

[21] Figure 9c illustrates the operation of a remote copy where null data is skipped by employing a message to skip it;

[22] Figure 9d illustrates skipping transmission and copying of null data;

[23] Figures 10a and 10b illustrate the relationship of operations at the primary storage system and the secondary storage system for one embodiment of the invention;

[24] Figure 11 shows an example of an "skip message" using a vendor-specific command;

[25] Figure 12 is another example of a skip message, but one which illustrates a data pattern;

[26] Figure 13 is a flow chart illustrating operations of the primary storage system and operation of the direct memory access unit to locate null data; and

[27] Figure 14 is a diagram illustrating various modes of operation of the storage system.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[28] Figure 1 is a diagram illustrating a typical prior art storage system in which a primary storage subsystem 102 is coupled to a secondary storage subsystem 103 through a network 105. Typically, network 105 will be a conventional network, for example, a network using ATM, sonet, dark fibre, or internet protocol communications. Furthermore, while Figure 1 depicts a single primary storage subsystem and a single secondary storage subsystem, as many or as few storage systems as desired may be employed, remotely situated

from one another in desired locations. Conventional uses for such storage systems are to facilitate sharing of data in a large enterprise, to provide increased reliability by providing backup copies of data, or the like.

[29] The primary storage system generally operates under control of a host 101 which is coupled to the storage subsystem 102 by a channel 103. Channel 103 interfaces to the storage subsystem 102 via a channel controller 111. Typically, another channel controller 112 is used to interface the storage subsystem 102 with a channel 104 to network 105.

[30] The secondary storage system 103 is similarly configured with host 106 controlling the secondary system 103 via channel 108 and channel controller 122. Another channel controller 121 provides an interface between the secondary storage subsystem 103 and the channel 109 coupled to network 105. Typical implementations for channels 103, 104, 108, and 109 are fibre channel, ESCON, SCSI, or GE. Channels 104 and 109 couple to network 105, which itself can be public or private.

[31] Storage system 102, in addition to the channel controllers already mentioned, includes a cache memory 110 operating under control of a control table 117. An internal bus 114 allows cache memory 110 to communicate with the channel controllers 111 and 112 and with a disk controller 113. Through another internal bus 115, for example a fibre channel, SCSI, or ATA bus, disk controller 113 communicates with storage volumes 116. The storage subsystems are generally configured similarly. Thus, in general, the structure of the primary storage subsystem 102 is matched by the structure of the secondary storage subsystem 103.

[32] Generally, data is shifted in and out of the primary storage subsystem via the channel controllers and the cache memory. As data arrives to be written into the storage volumes, it is temporarily stored in the cache memory and then retrieved by the disk controller to be written into the volumes. Similarly, as data is retrieved from the volumes 116 to be transmitted out over the network, it will generally be stored in the cache memory before being supplied to the channel controller in larger blocks. The storage subsystem translates physical disk addresses to logical disk addresses which are viewed by the host.

[33] As mentioned above, in a typical operation, it will be desired to "mirror" data from volumes 116 in the primary storage subsystem 102 onto volumes 126 in the secondary storage subsystem 103. As also mentioned above, because of the substantial storage capability typically provided in systems such as depicted in Figure 1, there can be substantial amounts of "null" data present on volumes 116. This null data can take the form of large portions of formatted, but unwritten, regions on the hard disk drives, or of files or portions of files for which directory entries are no longer valid (indicating erasure, etc.). To improve the

performance of the storage subsystems, reduce the load on the channel controllers and network, it is advantageous not to transmit this large quantity of null data through the primary storage subsystem, across channel 104, through network 105, and into channel 109 where it is rewritten onto disk volumes 126. Below we explain how this is achieved.

[34] Figure 2 is a block diagram of the channel controller, for example any of the channel controllers shown as blocks 111, 112, 121 or 122 in Figure 1. The channel controller 201 of Figure 2 includes within it a channel interface 204, a central processing unit 206, memory 205, a direct memory access circuit 207, and interface 208 to an internal bus 203. Interface 204 serves to interface the channel controller with higher level functionality, for example, a network or a host, while interface 208 provides an interface for channel controller 201 to internal functionality of the storage subsystem, such as a cache memory or disk controller. CPU 206 controls the components of the channel 201 by communications over bus 209. The channel interface 204 controls the channel protocol and controls the transfer of data to and from the channel, and with CPU 206 and DMA 207, the transfer of data between memory 205 and channel interface 204. The internal interface 208 controls the protocol of transmissions on internal bus 203, and the control of data over that line in response to activities of the DMA controller 207 and memory 205.

[35] Figure 3 is a schematic diagram illustrating the structure of the disk controller 301. Controller 301 can be used to implement disk controllers 113 or 123 in Figure 1. The disk controller has two types of interfaces. One interface, the internal interface 304, is an interface for the internal bus 302 (or 203 in Figure 2). The other interface 308 is an interface for disk bus 303 to enable communications to the storage volumes coupled to the disk bus 303. The internal interface 304 and the disk interface 308 are coupled via bus 309. The disk controller includes a CPU 306 and memory 305, as well as a DMA controller 307. These components regulate the flow of information between the internal bus 302 and the disk bus 303. The internal interface 304 controls the internal bus protocol and transfers data to and from the internal bus, and to and from memory 305, in conjunction with DMA controller 307. Similarly, the disk interface 308 controls the protocol on the disk bus 303 and transfers data under control of CPU 306, DMA controller 307 between the memory 305 and the disk bus 303. As mentioned above in some embodiments, disk controller 301 is capable of detecting null data in the storage volumes associated with the disk bus 303. DMA controller 307 employs known technology to "snoop" bus 309. When DMA controller 307 detects null data, it places appropriate signals on bus 309 to alert the other components coupled to the bus. This is discussed further below.

[36] Figures 4-6 illustrate an overview of an initial copy process on each of the primary and secondary storage subsystems. Figure 4 is a diagram which illustrates the primary storage subsystem 102 as it operates to make an initial copy. The first operation which occurs is that disk controller 113 reads disk data from a disk 116b and provides it to cache memory 110. The channel controller 112 operating under control of its internal CPU (not shown) monitors the transactions, and when an appropriate amount of data is stored into cache memory 110 along with associated entries of disk controller 113 into the control table 117, initiates a transfer. When this desired amount of data is stored into the cache memory 110, channel controller 112 issues a "write" command (or a vendor-specific command in the case of certain types of systems as discussed below) begins to process the initial copy. The channel controller 112 sends the initial copy data from cache memory 110 out over channel 104 to network 105. Once on network 105 the data is transmitted to the secondary storage system (not shown).

[37] Figure 5 illustrates the operations of the primary storage subsystem 102 during a remote copy process. The process is initiated by host 101 issuing a write command to the system 102. When this occurs, the channel controller 111 receives and analyzes the command. For this command, the channel controller 111 receives write data from host 101 which it then stores in cache memory 110. Controller 111 also stores information about this remote copy into control table 117. If the data from the cache memory 110 is ultimately stored on a disk as one of the pair of remote copy pairs, then channel controller 112 will issue a remote copy command to the secondary storage subsystem (not shown).

[38] Figure 6 is a diagram which illustrates the operation of the secondary storage subsystem 103 when an initial copy or remote copy data arrive over the network via channel 109. In this case the channel controller 121 analyzes the incoming command and stores the data into the cache memory 120 and makes an appropriate entry in the control table 127. The disk controller 123 periodically checks the control table 127. If the disk controller 123 finds procedures which are to be performed by the disk controller, then the disk controller reads data from cache memory 120 and writes it onto the target disk, in this case, disk 126b.

[39] Note that in a preferred embodiment of this invention, any overhead associated with the step designated (2) in Figure 4, (2) in Figure 5, or (3) in Figure 6 is reduced when null data is to be transferred. In other words, the operations discussed with respect to Figures 4-6 will be carried out just as described when ordinary (non-null) data is being processed. These procedures, however, will not be necessary when null data is being processed.

[40] Figure 7 is a flowchart illustrating a preferred embodiment of the method of this invention. The flowchart illustrates the initial copy procedure performed at the location of the primary storage subsystem 102 (see Figure 1). Before the initial copy procedure begins, however, the primary and secondary systems are synchronized to establish the remote copy pair, i.e., the correspondence between the disks at the primary storage subsystem and those at the secondary storage subsystem location. This activity is performed by a system administrator. After completion of that well known procedure, the initial copy procedure of Figure 7 is performed. The first step in carrying out this implementation of the initial copy procedure is cleaning of the file system 701. In this step, the unused portion of the primary disk, and all of the secondary disk, are filled with null data. This operation is performed using commercially available software, for example, BC Wipe™ available from Jetico of Tampere, Finland. The combination of using the cleanup software and skipping the copying of null data as described herein substantially reduces the overhead of the remote copy procedure. The cleanup procedure, however, is not mandatory, as will be evident from the following explanation.

[41] Once the cleanup is completed, the making of the initial copy begins with step 702 in which data is read from the target disk. In this operation, the disk controller 113 reads data from the disk and provides it to the cache memory 110. The reading and writing to the cache is usually performed using large amounts of data for each transfer, for example, 1 megabyte. The transfer itself is carried out under control of DMA 308.

[42] As shown by Figure 7, as the data is read, it is checked to determine whether null data is present at step 703. This operation is performed by DMA 307 monitoring bus traffic to determine the data pattern. Such an operation is usually termed "snooping." When DMA 307 detects that the data pattern consists of null data, that information is provided to CPU 306. The determination of whether data is null data or not will be based upon the particular software used for the cleaning operation, or the particular software used to format the hard disk before the initial copy is performed. Typically, the null data will consist of either all zeros, all ones, or a repeating pattern of zeros and ones, for example, 010101. When this pattern occurs, the CPU 306 instructs the system to skip copying that null data by using control table 117. On the other hand, when the data consists of random patterns of ones and zeros, then the CPU 306 places an entry in the control table 117 enabling normal operation.

[43] As the data is transferred, the channel controller 112 periodically checks the control table 117. When the channel controller, using table 117, determines that "normal" data is to be copied – the initial copy – then the channel controller 112 does a data transfer. During this transfer, data will ultimately be copied from the primary storage system 102 to the secondary storage system 103. The CPU 206 then sends commands to DMA 207 to transfer data from the cache memory 110 into the memory 205. After that, the CPU 206 sends a command to the channel interface unit 112 causing it to issue the initial copy (remote copy) command to the secondary storage system 103 over network 105. The issuance of the remote copy command is shown at step 704.

[44] When the request is to skip copying null data, CPU 206 issues a skip message, and does not transfer data. The skip message itself indicates that one or more quantities of data (the 1 megabyte size referred to above) are not to be copied. The disk controller continues to repeat steps 702-705 shown in Figure 7 until all of the data on the disk has been transmitted to the secondary subsystem or designated to be null data. This is shown by the test at step 706.

[45] As shown by step 707, once all of the data on a disk has been transferred or determined to be null data, the disk controller then selects the next disk or volume and repeats the procedure. Eventually, as shown by step 708, all of the volumes of storage at the primary location will have been initially copied or remote copied to the secondary storage location. At this point, the initial copy process is completed at the primary storage location 102.

[46] Figure 8 is a flowchart which illustrates the operation of the initial copy process at the secondary storage system 103. As described above, the initial optional step is for the administrator to format the volumes of the secondary storage, as shown by step 801. When a command to perform the initial copy at the secondary storage location is received at step 802, the procedure begins. At this time, the channel controller 121 receives the initial (remote) copy command over network 105. The CPU 206 analyzes the command which is received by the channel interface 204. If the command is to copy normal data, then the CPU 206 commands that channel interface 204 to receive data from the primary storage system 102 via network 105. As the data arrives, the channel interface 204 stores the data into memory 205.

[47] Next, under control the CPU 206, the DMA unit 207 is instructed to move data from memory 205 to cache memory 120. This request from CPU 206 is stored in control table 127. The disk controller 123 periodically checks the control table 127. When the disk controller 123 detects a request, the CPU 306 commands the disk interface 308 to issue a

write command and write the data into the disk. This is shown in the flowchart by steps 803, 804, and 806. This is the operation for normal (non-null) data.

[48] On the other hand, if the command received by channel controller 121 is a skip message indicative of null data, then the CPU 206 begins the skip procedure. Thus, null data has been detected at step 803, and the skip null data operation of step 805 is performed. In this situation, the secondary storage system 103 does not copy the data onto the disk, because the disk, by previously having been cleaned or formatted, has that null data already stored in the addressed location. If the skip message is to write data of some pattern, and that pattern is not already in place in the secondary storage location because the disk was not cleaned or formatted, then the disk controller 123, when it detects the skip request, will write the specified pattern onto the disk. This is achieved by having CPU 306 generate the desired pattern in memory 305 and provide commands to the disk interface unit 308 to write the data into the disk. The procedure described above, as shown by step 807, is carried out repeatedly until the initial copy at the secondary storage location 103 is completed.

[49] As described earlier, there are several methods for skipping null data during reading and writing in the secondary storage location. Figures 9b and 9c illustrate two of these methods of skipping null data. Figure 9a illustrates the procedure if null data is not skipped. Figure 9a is discussed first. In Figure 9a Data 0 and Data 3 are normal data, while Data 1 and Data 2 are null data. In the normal initial or remote copy operation as shown by Figure 9a, the primary storage subsystem 102 will issue four copy commands, one for each of the sets of data, Data 0, Data 1, Data 2 and Data 3. These commands cause the data to be transferred from the memory in the primary storage system 102 to the memory in the secondary storage system 103. Once in the secondary storage system, four more commands are decoded and carried out, resulting in the writing of Data 0, Data 1, Data 2 and Data 3 onto the disks of the secondary storage system. As evident from the discussion above, essentially all of the time and communication bandwidth used to transfer Data 1 and Data 2 within the primary storage system, across the network and then to rewrite it into the secondary storage system are unnecessary and wasted.

[50] Figures 9b and 9c illustrates two techniques for skipping null data. Figure 9b illustrates the skipping of null data in the situation in which the secondary storage subsystem volumes have already been formatted in a manner to match the formatting of the null data in the primary storage system. In the case of Figure 9b, using the techniques described above in conjunction with Figures 7 and 8, the primary storage system detects that Data 1 and Data 2 are null data, and that there is no need to transmit them to the secondary storage system.

Knowing that the secondary storage system is already appropriately formatted, the primary storage system does not need to transmit Data 1 or Data 2, or otherwise send any indication of the existence of the Data 1 or Data 2. Once Data 0 and Data 3 are copied into the secondary storage subsystem volumes, the contents of those volumes will match (mirror) the contents of the corresponding volumes in the primary storage system 102.

[51] An alternative procedure may be implemented when the secondary storage subsystem 103 is not formatted. In this circumstance, if the primary storage system issues the initial copy command sequentially, the secondary storage system will be able to detect that the addresses for Data 1 and Data 2 have been skipped. In this circumstance, while null Data 1 and null Data 2 are not transmitted across the network, the secondary storage system writes the appropriate form of null data onto its storage volumes because it is able to detect the skipped addresses of Data 1 and Data 2.

[52] Figure 9c illustrates another implementation of a preferred embodiment of this invention. In the case of Figure 9c, the skipping of null data is indicated by the transmission of a message from the primary storage subsystem 102 to the secondary storage system 103 indicating that null data has been skipped. In this circumstance the volumes in the secondary storage subsystem do not need to be formatted in advance. As shown by the diagram, the existence of null data as consisting of Data 1 and Data 2 is detected by the primary storage subsystem, for example by the DMA unit snooping the bus on which the data is available. When some of the data, for example Data 1 and Data 2, are determined to be null data, the primary subsystem 102 bundles them together into one message and sends that information to the secondary storage subsystem 103. The secondary storage subsystem, having analyzed the skip message sent to it from the primary storage subsystem 102, proceeds to write null data into the storage volumes in the location specified by the skip message for the locations of Data 1 and Data 2. The transmission of the skip message, although it does not eliminate the need to write the appropriate data into the secondary storage system, does lessen traffic on the network, and reduce overhead at the secondary location, yielding additional bandwidth for the transfer of "real" data between the primary and secondary subsystems.

[53] Figure 9d illustrates the operation of skipping null data from another perspective. In the upper portion of the figure, the data (null and non-null) is shown along with the normal commands that would write it onto the secondary system. The command (Cmd) is to write the block of data of the designated size and address (Adr). For example, the first command is to write data of a size 0x0010 at address 0x0030. The lower portion of the figure illustrates how only the non-null data is transmitted to the secondary subsystem.

[54] Figure 10a is a diagram illustrating the skip message. On the left-hand side of Figure 10a, note the number of transfers of data between the primary and the second storage system for the purpose of transmitting null data from one to the other. On the right-hand side of Figure 10a, a skip message is used to indicate that the designated data "chunks" may be skipped. All of the steps of transmitting data (FCP_DATA) are eliminated.

[55] Figure 10b is a diagram similar to Figure 10a. As shown in the left-hand side of Figure 10b, if the null data is transmitted, numerous chunks of null data are transferred from the primary to the secondary storage system. In contrast, on the right-hand side of Figure 10b, implementation of the method of this invention has eliminated the need to transmit all of the null data between the two storage subsystem. The null data transfer is shown by the same command as described in conjunction with Figure 10a.

[56] Figure 11 is an example of a skip message. The message consists of nine bytes (shown in the rows) with the state of each bit of the nine bytes shown in the columns. The operation code indicates to skip copying, with the starting logical block address shown in bytes 2-5 and the transfer length in bytes 7-8.

[57] Figure 12 is an example of a skip message, but which includes a desired data pattern to be written at the secondary storage subsystem location. The skip message is of the same format as Figure 11, but the desired data pattern to be written on the storage volumes is provided as byte 6 of the message. Of course, the message format shown in Figures 11 and 12 are exemplary. Vendor-specific messages may be employed in which various sizes of null data are to be skipped based on a particular message or command. For example, a specific message can be sent to skip copying the next 10 bytes, the next 100 bytes, etc.

[58] Figure 13 is a flowchart illustrating the procedure followed when host writes are performed using a channel controller to the primary storage subsystem. In this circumstance, the initial storage of data in the primary system can be expedited when the write operations detect null data. In this case, the CPU 206 of the channel controller 111 handles host input/output operations, that is, steps 1301 and 1303. The CPU 206 of channel controller 112 handles the remote copy procedure, that is, steps 1303, 1304 and 1305.

[59] When the host 101 issues a "write" command, the channel interface 204 receives it and reports it to the CPU 206, sometimes using memory 205 to store it. CPU 206 analyzes the command, and if it is valid begins to execute it. If the command is invalid, CPU 206 issues a "no good" status via the channel interface 204 back to host 101. Such an operation is, for example, in accordance with the SCSI specification.

[60] Assuming the command is valid, CPU 206 instructs the interface unit 204 to begin transferring data from host 101. Interface unit 204 transfers data from the channel 103 to the memory 205. The CPU 206 waits until a specified amount of data is transferred into memory 205, and then issues a command to DMA 207. DMA 207 is instructed to transfer data from memory 205 to cache memory 110 using the internal interface 208 and the internal bus 203. The CPU 206 controls the interface 208 so that it does not overtake the channel interface 204. When all of the data is transferred into the cache memory 110, the CPU 206 sends a "good" status report back to host 101 via channel interface 204 and channel 103. As these procedures are carried out, the CPU 206 will periodically update the control table 117. All of these operations are subsumed within the right command step 1301 of Figure 13.

[61] As the transfer is occurring in the manner described above, the DMA 207 will snoop on the data pattern. If the data is null data, the DMA 207 will cause the CPU 206 to put a skip message into control table 117. The CPU 306 of the disk controller 113, as it writes data, periodically checks the control table 117. When that CPU 306 detects a skip message, which will be processed by the disk controller 113, CPU 306 begins the procedure to write the data. All of the information needed for the procedure is now included within control table 117. CPU 306 commands the DMA 307 to transfer data from the cache memory 110 to the memory 305 at a pace which does not overtake channel controller 111 checking with control table 117. During this data transfer, the DMA 307 can also detect null data. CPU 306 waits until a specified amount of data are transferred into memory 305. If the data is null data, the CPU 306 puts a skip message into the control table 117. CPU 306 then begins the data transfer from memory 305 to disk 115 via the disk interface unit 308 and the disk bus 303. CPU 306 controls the disk interface 308 to assure that it does not overtake the internal interface 304. Once all of the data is written onto disk 115, the CPU 306 issues a remote copy information into control table 117, again as shown by step 1302.

[62] Channel controller 112 periodically checks control table 117. When the channel controller 112 detects a remote copy request, it starts the remote copy procedure. If there are two successive remote copy procedures, the CPU 206 may combine them into one remote copy command, either generic or vendor specific. Before issuing the remote copy command, the CPU 206 checks the control table 117. If it finds a skip message, then the CPU 206 will issue that skip message as shown by steps 1305. If CPU 206 does not find a skip message, then it issues the remote copy command and transfers data to the secondary storage subsystem 103. The remote copy operation at the secondary storage subsystem 103 is the same as described in conjunction with steps 802-806 of Figure 8.

[63] Figure 14 is a diagram which summarizes different operations of the methods and apparatus described above. In Figure 14 the first column indicates the particular application, in other words whether null data is detected during the making of an initial copy at the secondary location, or even earlier during the write to the primary storage location. The next column of Figure 14 identifies what component detects the null data at the primary storage location. In the case of the initial copy from the primary storage subsystem to the secondary storage subsystem, the null data is detected using the disk controller. In contrast, in the case of the data first being written into the primary storage system from the host, the channel controller detects the null data.

[64] The next column of the chart summarizes what occurs in response to the detection of null data. In the case of the detection occurring at the disk controller, i.e., based upon the initial copy operation, either a skip message can be generated, or the data can be ignored. These two conditions were described in conjunction with Figures 9b and 9c respectively. If the channel controller detects the null data as it arrives from the host, a skip message is employed to convey to the downstream disks that writing of that data is to be skipped. This condition was explained in conjunction with Figure 9c.

[65] The right-hand column of Figure 14 illustrates what happens at the target subsystem. In the case of the initial copy application, all of the actions at the target subsystem occur at the secondary system. In the case of the skip message, if the disk has already been appropriately formatted, no action is necessary at the secondary subsystem. If the disk has not been previously formatted, then as shown in the right-hand block, no data needs to be written. Similarly, for the situation in which the initial copy is performed, and the null data is to be just skipped, if the disk has been previously formatted, no action is necessary at the secondary subsystem. If the disk is not, has not been previously cleaned or formatted, null data must be written. The bottom row of the table in Figure 14 illustrates the operation for the host write application. In this circumstance, a skip message will result in the writing of null data at the primary subsystem.

[66] The foregoing has been a description of the preferred embodiment of the invention. It will be appreciated that variations may be made to implement the invention without departing from the scope of the following claims.

What is claimed is:

1. A method for copying information from a primary storage system to a secondary storage system, the method comprising:

at the primary storage system, analyzing the information to be copied to determine if it constitutes data desired to be copied or null data which does not need to be copied;

transmitting to the secondary storage system the data desired to be copied and not transmitting the null data;

at the secondary storage system, writing to a storage unit in the secondary storage system the information to be copied.

2. A method as in claim 1 wherein before the step of at the secondary storage system copying to a storage unit, a step is performed of formatting the storage unit to contain null data.

3. A method as in claim 1 further including the step of transmitting to the secondary storage system a skip message which indicates that a block of null data is present on the primary storage system and will not be transmitted to the secondary storage system.

4. A method as in claim 3 wherein after receiving the skip message the secondary storage system writes null data onto the storage unit.

5. A method as in claim 1 wherein the step of transmitting further comprises transmitting to the secondary storage system a pattern of null data to be written onto the secondary storage system at the location where null data would have been stored had it been transmitted to the secondary storage system.

6. A method as in claim 3 wherein the skip message further comprises a pattern for data to be written onto the secondary storage system at the location where null data would have been stored had it been transmitted to the secondary storage system.

7. A method as in claim 1 wherein the primary storage system includes a disk controller and the disk controller performs the step of analyzing the information to be copied to determine if it constitutes data desired to be copied or null data.

8. A method as in claim 1 wherein:

the data to be copied is transmitted from the primary storage system to the secondary storage system in a sequential order; and

the secondary storage system writes null data onto the storage unit when the secondary storage unit detects gaps in the sequential order.

9. A method as in claim 1 wherein the step of analyzing the information is performed as the information is transmitted on a bus in the primary storage system.

10. A method as in claim 9 wherein a DMA controller is used to snoop the bus and analyze the information.

11. In a system having a host and a storage system coupled to the host, the storage system including a channel controller coupled to the host and apparatus for storing data, a method for writing data into the apparatus comprising:

monitoring the data as it is transferred to the storage system;

when a block of data consists only of null data sending a skip message to the apparatus to cause the apparatus not to write that null data; and

when a block of data consists of non-null data sending that data to the apparatus and causing the apparatus to write that data.

12. A method as in claim 11 wherein before the non-null data is written, a step is performed of formatting the apparatus to contain null data.

13. A method as in claim 11 wherein the skip message indicates the address of the null data.

14. A method as in claim 13 wherein the skip message also indicates an amount of null data.

15. A method as in claim 13 wherein after receiving the skip message the apparatus writes null data at the address of the null data.

16. A method as in claim 11 wherein the skip message further comprises a pattern for the null data and the address of the null data.

17. A method as in claim 16 wherein after receiving the skip message the apparatus writes the pattern for the null data at the address of the null data.

18. A method as in claim 11 wherein the storage subsystem includes a disk controller and the disk controller performs the step of analyzing the information to be copied to determine if it constitutes null data.

19. In a storage system operating under computer control and coupled to receive information to be stored as well as specified addresses at which the information is to be stored, the information including null data and non-null data, a method comprising:

monitoring the specified addresses to determine if there are any missing addresses in the addresses received; and

writing null data at the missing addresses.

20. A method as in claim 19 further comprising writing the non-null data at the specified addresses.

21. At a primary storage system, a method of selecting stored data to send from the primary storage system to a secondary storage system comprising:

reading a selected amount of the stored data;

determining if the stored data is null data or non-null;

if the stored data is non-null data then sending it with a command to the secondary storage system to cause the secondary storage system to store the data;

if the stored data is null data then not sending it to the secondary storage system; and

repeating the preceding steps until all of the stored data has been either sent to the secondary system or has been determined to be null data.

22. A method as in claim 21 wherein the step of if the stored data is null data then not sending it to the secondary storage system further comprises sending to the secondary storage system a message indicating that null data has not been sent.

23. A method as in claim 22 wherein the message indicates an address of the null data and a quantity of the null data.

24. At a secondary storage system coupled to receive data transmitted from a primary storage system, a method of storing data transmitted from the primary system comprising:

receiving a command from the primary storage system to store data that will be transmitted from the primary storage system;

receiving the transmitted data from the primary storage system;

receiving a series of addresses, each for a selected amount of the transmitted data;

analyzing the series of addresses to determine if addresses have been skipped in a sequence of addresses;

storing the data at locations specified by the addresses;

storing null data at addresses that have been skipped in the series of addresses;

repeating the preceding steps until all of the data transmitted has been stored.

25. A method as in claim 24 wherein the step of storing null data at addresses that have been skipped in the series of addresses is performed by formatting the secondary storage system before the step of storing the data at locations specified by the addresses.

26. A method as in claim 24 wherein the step of storing null data at addresses that have been skipped in the series of addresses is performed by writing a selected data pattern at addresses that have been skipped in the series of addresses.

Method and Apparatus for Skipping Remote Copy of Null Data in Storage Systems

ABSTRACT OF THE DISCLOSURE

A method and apparatus are provided for enhancing the performance of storage systems is described. In the making of an initial copy to a secondary subsystem, or in the initial storage of data onto a primary storage subsystem, null data is skipped. The data may be skipped by sending the non-null data in sequence so missing addresses are identified as being null data, or a skip message may be used to designate regions where null data is to be present.

【図１】

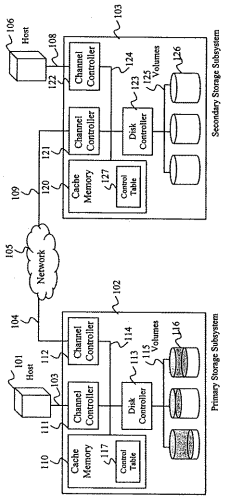


Figure 1 (Prior Art)

【図２】

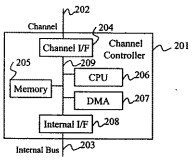


Figure 2

【図３】

【図４】

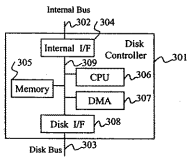


Figure 3

【図５】

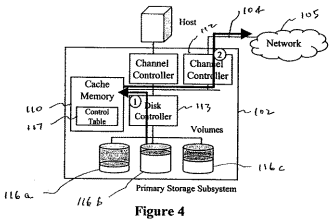


Figure 4

【図６】

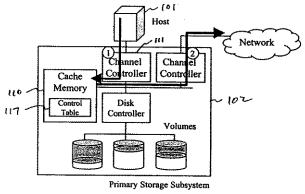


Figure 5

【図７】

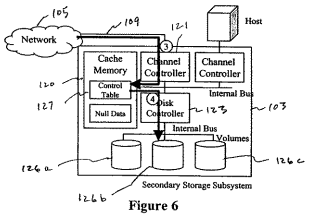


Figure 6

【図８】

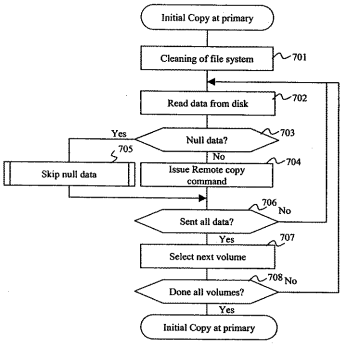


Figure 7

【図９Ａ】

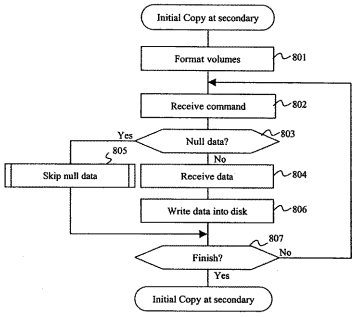


Figure 8

【図９Ｂ】

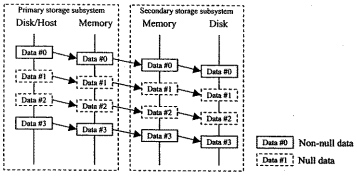


Figure 9(a) Remote Copy - Null Data Not Skipped (Prior Art)

【図９Ｃ】

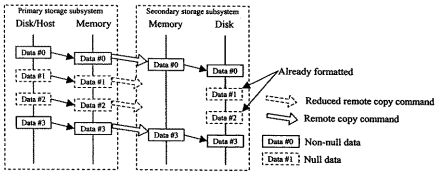


Figure 9(b) Remote Copy with Null Data Skip

【図９Ｄ】

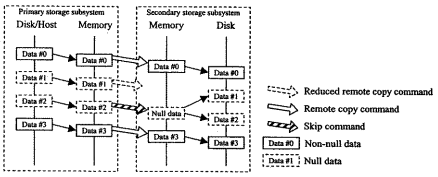


Figure 9(c) Remote Copy with Null Data Skip Using Skip Message

【図１０Ａ】

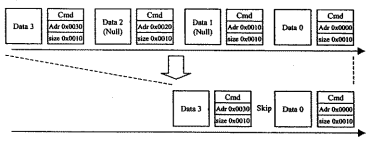


Figure 9(d) Remote Copy Showing Null Data Skip

【図１０Ｂ】

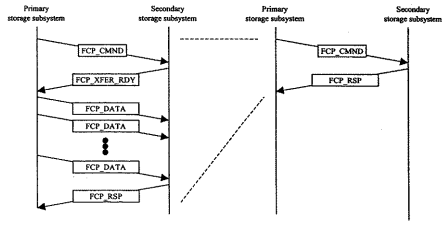


Figure 10(a) Skip Procedure for Null Data

【図１１】

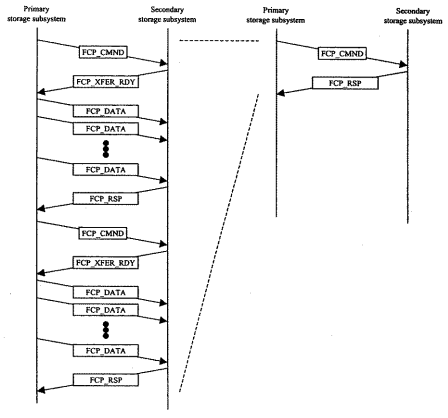


Figure 10(b) Skip Procedure for Successive Null Data

【図１２】

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (0xCA)							
1	Reserved			DPO	FUA	Reserved		RELADR
2	(MSB)							
3				LOGICAL BLOCK ADDRESS				
4								
5								(LSB)
6	Reserved							
7	(MSB)							
8				TRANSFER LENGTH				(LSB)
9	Control							

Figure 11

【図１３】

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (0xCA)							
1	Reserved			DPO	FUA	Reserved		RELADR
2	(MSB)							
3				LOGICAL BLOCK ADDRESS				
4								
5								(LSB)
6	Pattern							
7	(MSB)							
8				TRANSFER LENGTH				(LSB)
9	Control							

Figure 12

【図１４】

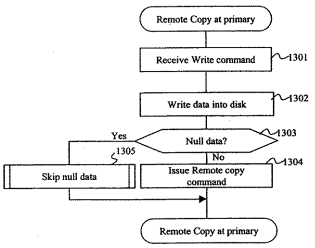


Figure 13

#	Application	Detect null data at primary	null data information	Action at target subsystem
1	Initial copy	Disk controller	Skip message	Write null data
2				Nothing (formatted)
3			Just skip	Write null data
4				Nothing (formatted)
5	Host write	Channel controller	Skip message	Write null data at primary

Figure 14