(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4969791号
(P4969791)
(45)発行日　平成24年7月4日(2012.7.4)　(24)登録日　平成24年4月13日(2012.4.13)

(51)Int.Cl.　F I
G06F 3/06 (2006.01)　G06F 3/06 304N
G06F 3/06 540

請求項の数 10　(全 21 頁)

(21)出願番号　特願2005-97505(P2005-97505)
(22)出願日　平成17年3月30日(2005.3.30)
(65)公開番号　特開2006-277487(P2006-277487A)
(43)公開日　平成18年10月12日(2006.10.12)
審査請求日　平成19年11月26日(2007.11.26)

(73)特許権者　000005108
株式会社日立製作所
東京都千代田区丸の内一丁目６番６号
(74)代理人　110000062
特許業務法人第一国際特許事務所
(72)発明者　新井　政弘
神奈川県川崎市麻生区王禅寺１０９９番地
株式会社　日立製作所　システム開発研究所内
(72)発明者　松並　直人
神奈川県川崎市麻生区王禅寺１０９９番地
株式会社　日立製作所　システム開発研究所内

審査官　菅原　浩二

最終頁に続く

(54)【発明の名称】ディスクアレイ装置およびその制御方法

(57)【特許請求の範囲】
【請求項１】
ディスクアレイコントローラと前記コントローラを制御する制御プログラムを備えるディスクアレイ装置であって、
前記ディスクアレイコントローラは、物理的に分離不可能な複数個のプロセッサコアを内蔵したＣＰＵを搭載しており、
前記制御プログラムは、各々処理完結動作するプログラムモジュールと、前記プロセッサコアを管理する情報を有し、該管理情報を用いて、該プロセッサの状態を管理する管理部とを含み、
前記各プロセッサコアはそれぞれ独立した単位プロセッサであり、
前記単位プロセッサには、前記各々完結したプログラムモジュールが動的または固定、半固定の何れかになるように割り当てられ、
該割り当てられたプログラムモジュールの１つが、前記管理部の管理情報を用いて自己および他の単位プロセッサの動作負荷または動作状態を監視し、該単位プロセッサの負荷分散が必要であると判断したとき、前記プログラムモジュール群動作中に該負荷分散が必要である単位プロセッサのプログラムモジュールを未割当単位プロセッサに動的に割り当て、負荷分散処理することを特徴とするディスクアレイ装置。
【請求項２】
請求項１に記載のディスクアレイ装置において、
前記割り当てられたプログラムモジュールの１つが、前記単位プロセッサの動作負荷ま

たは動作状態を監視する監視制御プログラムモジュールであり、該監視制御プログラムモジュールは、固定または半固定になるように割り当てた単位プロセッサの動作負荷または動作状態が、前記管理部の管理情報に基づいて負荷分散開始を決定する規定値に達すると、該単位プロセッサへのプログラムモジュール割り当てを維持しつつ、該単位プロセッサとは別の単位プロセッサに前記プログラムモジュールを動的となるよう追加して割り当てることを特徴とするディスクアレイ装置。

【請求項３】

請求項１に記載のディスクアレイ装置において、

前記単位プロセッサは、固定または半固定になるように前記プログラムモジュールを割り当てた単位プロセッサの動作負荷が相対的に高いとき、動的になるように割り当てたプログラムモジュールの動作負荷が相対的に低い単位プロセッサの処理を終了させ、終了させた単位プロセッサに、前記相対的に動作負荷が高い単位プロセッサに割り当てたプログラムモジュールを、動的になるように割り当てることを特徴とするディスクアレイ装置。

【請求項４】

請求項３に記載のディスクアレイ装置において、

前記単位プロセッサは、前記プログラムモジュールの動作負荷が相対的に高い単位プロセッサがグループ化され、該グループ化された同一のグループ内に属する単位プロセッサで、かつ、単位プロセッサが動的になるように割り当てたプログラムモジュールの動作負荷が相対的に低い単位プロセッサの処理を終了させ、該終了した単位プロセッサに、前記相対的に動作負荷が高い単位プロセッサに割り当てたプログラムモジュールを、動的になるように割り当てることを特徴とするディスクアレイ装置。

【請求項５】

請求項３に記載のディスクアレイ装置において、

前記単位プロセッサは、どのグループにも属さない単位プロセッサに、前記相対的に動作負荷が高い単位プロセッサに割り当てた制御プログラムモジュールを、動的になるよう割り当てるとともに、割り当てた単位プロセッサを、前記相対的に動作負荷が高い単位プロセッサが属するグループに属するようにすることを特徴とするディスクアレイ装置。

【請求項６】

ディスクアレイコントローラと前記コントローラを制御する制御プログラムを備えるディスクアレイ装置であって、

前記ディスクアレイコントローラは、物理的に分離不可能な複数個のプロセッサコアを内蔵したＣＰＵを搭載しており、各プロセッサコアはそれぞれ独立した単位プロセッサとなり、前記制御プログラムは、各々処理完結動作するプログラムモジュール群と、前記プロセッサコアを管理する情報を有し、該管理情報を用いて、該プロセッサの状態を管理する管理部を含み、

前記単位プロセッサには、前記各々処理完結したプログラムモジュールが動的または固定、半固定の何れかになるように割り当てられ、該制御プログラムの１つである監視制御プログラムモジュールは、前記管理部の管理情報を用いて自己および他の単位プロセッサの動作負荷または動作状態を監視し、特定のプロセッサコアに障害が発生した場合に、前記プログラムモジュール群間での制御、非制御の関係を表す特権レベルの低い制御プログラムモジュールから順にすべて終了させることを特徴とするディスクアレイ装置。

【請求項７】

請求項６に記載のディスクアレイ装置において、

前記単位プロセッサは、前記障害が発生したプロセッサコアの単位プロセッサに割り当てたプログラムモジュールの割り当て形式が半固定であり、かつ、前記プログラムモジュールの割り当て形式を動的とした別の単位プロセッサがあるとき、該単位プロセッサの割り当て形式を動的から半固定に変更することを特徴とするディスクアレイ装置。

【請求項８】

請求項６に記載のディスクアレイ装置において、

前記単位プロセッサは、前記障害が発生したプロセッサコアの単位プロセッサに前記プ

ログラムモジュールの割り当て形式が固定であるとき、他系のディスクアレイコントローラに処理引き継ぎを依頼し、他のプログラムモジュールの終了処理を省略する障害処理を行うことを特徴とするディスクアレイ装置。

【請求項９】

請求項１に記載のディスクアレイ装置において、

前記ディスクアレイコントローラは、物理的に分離不可能な複数個のプロセッサコアを内蔵したＣＰＵを搭載しており、内部スイッチ、不揮発性メモリコントローラ、揮発性メモリコントローラ、ディスクアレイコントローラ間転送コントローラ、パリティ演算器およびＣＰＵ内部キャッシュを有し、前記制御プログラムは、監視制御プログラムモジュール、運用管理プログラムモジュール、ＲＡＩＤ制御プログラムモジュール、ＮＡＳ制御プログラムモジュール、ホストＩ／Ｏ制御プログラムモジュール、ドライブＩ／Ｏ制御プログラムモジュール、初期ＭＰＵコア割当管理テーブル、初期閾値管理テーブル、ＲＡＩＤ設定管理テーブル、ＬＵ設定管理テーブル、およびＮＡＳボリューム管理テーブルを備えており、前記単位プロセッサは、自己および他の単位プロセッサの制御プログラムの負荷状態を監視し、前記管理部の管理情報である負荷閾値および閾値オーバ回数が規定回数を超えると、同一グループ内の未割当コアを検索し、検出した未割当コアに、前記プログラムモジュールを割り当てることを特徴とするディスクアレイ装置。

【請求項１０】

物理的に分離不可能な複数個のプロセッサコアを内蔵したＣＰＵを搭載し、各プロセッサコアはそれぞれ単位プロセッサとなるディスクアレイコントローラと、前記コントローラを制御し、各々処理完結動作するプログラムモジュールと、前記プロセッサコアを管理する情報を有し、該管理情報を用いて、該プロセッサコアの状態を管理する管理部を含む制御プログラムを備えるディスクアレイ装置の制御方法であって、

前記単位プロセッサに、前記各々完結したプログラムモジュールが動的または固定、半固定の何れかになるように割り当て、

前記割り当てた前記プログラムモジュールの１つが、前記管理部の管理情報を用いて自己および他の単位プロセッサの動作負荷または動作状態を監視し、

該単位プロセッサの負荷分散が必要であると判断したとき、前記プログラムモジュール群動作中に該負荷分散が必要である単位プロセッサのプログラムモジュールを未割当単位プロセッサに動的に割り当て、負荷分散処理することを特徴とするディスクアレイ装置の制御方法。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、複数のディスク装置を備えるディスクアレイ装置、およびディスクアレイ装置の制御方法に関する。

【背景技術】

【０００２】

複数のディスク装置を備えることによってデータに対するアクセス速度、ディスク装置に対する信頼性を向上させる技術として、ＲＡＩＤ(Redundant Array of Independent Disks)が知られている。一般にディスクアレイ装置では、このＲＡＩＤ制御を装置内に備えるディスクアレイコントローラが担っており、従来より処理性能向上のためディスクアレイコントローラの高性能化が図られてきた。

【０００３】

ディスクアレイコントローラにはアレイコントローラの処理プログラムを動作させるためのＣＰＵが搭載されている。高性能化の一手法として、ＣＰＵを複数乗せることによって処理を並列化し、高速化する手法が考えられる。

【特許文献１】特開平９－１６０８８９号公報

【特許文献２】特開平６－３５８７１号公報

【発明の開示】
【発明が解決しようとする課題】
【０００４】
しかしながら、複数ＣＰＵを搭載したディスクアレイ装置では、ＣＰＵ間での情報共有のために高価な共有メモリが必要となる。このため性能の向上分に対する相対的なコストが高くなってしまいコストパフォーマンスが悪化してしまうという問題がある。また、ＣＰＵ個別にクロックや電源供給回路が必要となるため、実装面でも肥大化し、結果として製造コストも増加してしまうといった問題がある。
【０００５】
一方、制御面で見ると、特許文献１では処理をジョブ単位で制御プロセッサないしプロセッサ群に割当てる方法が取られているが、この場合、複数の分割された制御（ジョブ）が1つのプロセッサないしプロセッサ群内に混在して存在するため、例えばディスクアレイ装置の運用管理機能がセキュリティクラック等の攻撃を受けてダウンしてしまうような場合、それにひきづられて他の制御もダウンしてしまう可能性がある。
【０００６】
本発明は、上記に示した課題を解決するためになされたものであり、高コストパフォーマンスの実現、他制御への影響抑止、リソースの有効活用を可能とするディスクアレイを提供することを目的とする。
【課題を解決するための手段】
【０００７】
上記課題を解決するために、本発明の実施形態では、複数プロセッサコア(ＭＰＵコア)を内蔵したＣＰＵを搭載するディスクアレイコントローラを備え、処理完結しているプログラムモジュールを各コアに静的または動的に割り当てて処理動作するディスクアレイ装置を提供する。ここで、処理完結しているプログラムモジュールとは、同一プログラム内で起動・停止・障害復旧処理及び外部への通知のための論理インターフェースを備え、他のプログラムと連携非同期で動作可能なプログラムを指す。
【０００８】
本発明の実施形態に係るディスクアレイ装置は、ＭＰＵコアを固定資源または動的資源として管理する手段と、各コアの負荷状況を監視し、必要に応じて外部からの指示なしに各制御へのコアの割当て数を変動させる監視制御手段と、コアに障害が発生した際に動作中プログラム間の依存関係を考慮して終了させ、必要に応じて他系のディスクコントローラにフェイルオーバする障害処理手段とを備えることを特徴とする。
【０００９】
すなわち、本発明は、ディスクアレイコントローラを備えるディスクアレイ装置であって、前記ディスクアレイコントローラは、物理的に分離不可能な複数個のプロセッサコアを内蔵したＣＰＵを搭載しており、各プロセッサコアはそれぞれ単位プロセッサとなり、一の単位プロセッサは、自己及び他の単位プロセッサを個別に管理し、各々完結した制御プログラムを、自己又は他の単位プロセッサにプロセッサ単位で割当てる際に、適宜動作終了可能とする動的になるように、又は前記ＣＰＵ全体の動作終了まで動作可能とする非動的になるように割り当て、割り当てた制御プログラムの動作負荷又は動作状態を単位プロセッサごとに管理するディスクアレイ装置である。
【発明を実施するための最良の形態】
【００１０】
本発明を実施するための最良の形態を説明する。
以下、本発明に係るディスクアレイ装置およびディスクアレイ装置の制御方法について図面を参照しつつ、実施例に基づいて説明する。
【００１１】
実施例１を説明する。本実施例のディスクアレイ装置の構成について、図１～図８を参照して説明する。図１は本実施例に係るディスクアレイ装置の概略構成を示す説明図である。図２は本実施例に係るディスクアレイ装置の概観図である。図３は本実施例に係るデ

ィスクアレイ装置におけるディスクアレイコントローラの機能的構成を示すブロック図であり、図４はその概観図を表している。図５は本実施例に係るディスクアレイ装置においてディスクアレイコントローラに搭載される複数プロセッサコア（ＭＰＵコア）を内蔵したＣＰＵ（以下、「マルチコアＣＰＵ」と記す）の機能的構成を示すブロック図である。

【００１２】

また、図６は制御プログラムに含まれる各プログラムモジュール及び管理テーブルを示す説明図であり、図７、８は前記管理テーブルのうちの２つを具体的に示した一例である。以下、順に説明を行う。

【００１３】

本実施例におけるディスクアレイ装置１は、ディスクアレイコントローラ１１、１２、接続インターフェース１３０、１３１、１３２、電源１０５、１０６、および、複数のディスク装置Ｄ００～Ｄ２Ｎを備えている。複数のディスク装置Ｄ００～Ｄ２Ｎは、例えば、図２に示すようにしてディスク装置１に備えられると共に、ＲＡＩＤシステムを構成している。

【００１４】

ディスクアレイコントローラ１１、１２は、制御プログラムを実行することによって、ディスクアレイ装置１における各種制御処理を実行する制御回路である。本実施例においては、２つのディスクアレイコントローラ１１、１２が備えられているが、１つまたは３つ以上のディスクアレイコントローラが備えられていてもよい。ディスクアレイコントローラ１１、１２は信号線１０１を介して相互に通信可能に接続されている。ディスクアレイコントローラ１１、１２はまた、ストレージネットワーク４０を介して各ホスト２０、２１、２２と接続され、管理用ネットワーク３０とを介して管理用端末装置３１と接続されている。ストレージネットワークは、例えば、ファイバチャネルによるＦＣ－ＳＡＮ（Storage Area Network）やＴＣＰ／ＩＰネットワークを利用したＩＰ－ＳＡＮなどであり、管理用ネットワークはＴＣＰ／ＩＰネットワークを利用したＬＡＮや、シリアルケーブルによるＰｏｉｎｔ ｔｏ ＰｏｉｎｔネットワークＰｏｉｎｔである。

【００１５】

ディスクアレイコントローラ１１、１２は、接続インターフェース１３０、１３１、１３２を介して複数のディスク装置Ｄ００～Ｄ２Ｎと接続されている。より具体的には、接続インターフェース１３０は、ディスクアレイコントローラ１１，１２と信号線１０２を介して直接接続されており、定期的な通信を行っている。また、各接続インターフェース１３０、１３１、１３２は互いに信号線１０３を介して接続されている。従って、接続インターフェース１３１は接続インターフェース１３０を介して、接続インターフェース１３２は接続インターフェース１３０、１３１を介してディスクアレイコントローラ１１、１２と接続されている。

【００１６】

接続インターフェース１３０は、複数のディスク装置Ｄ００～Ｄ０Ｎと接続され、接続インターフェース１３１は複数のディスク装置Ｄ１０～Ｄ１Ｎと接続され、接続インターフェース１３２は複数のディスク装置Ｄ２０～Ｄ２Ｎと接続されている。

【００１７】

ディスクアレイコントローラ１１、１２を含む接続インターフェース１３０及び複数のディスク装置Ｄ００～Ｄ０Ｎのグループは、例えば、基本筐体と呼ばれ、接続インターフェース１３１及び複数のディスク装置Ｄ１０～Ｄ１Ｎのグループ、及び接続インターフェース１３２及び複数のディスク装置Ｄ２０～Ｄ２Ｎのグループは、例えば、増設筐体と呼ばれる。なお、図１からも明らかなように、増設筐体は０ないし１つであってもよく、あるいは、３つ以上あってもよい。

【００１８】

なお、本実施例では基本筐体をディスクアレイコントローラ１１、１２および接続インターフェース１３０、複数のディスク装置Ｄ００～Ｄ０Ｎから成るグループとして記載しているが、基本筐体に複数のディスク装置Ｄ００～Ｄ０Ｎを含まない形態でも良い。

【００１９】
ホスト２０、２１、２２は、例えば、各種データを入力する端末装置であり、ホスト２０、２１、２２において処理されたデータは、逐次、ディスクアレイ装置１に対し送出され、ディスクアレイ装置１に格納される。なお、ホスト２０～２１は、１つであっても良く、あるいは、４つ以上備えられてもよい。
【００２０】
電源１０５は、電力線１０７及び接続インターフェース１３０を介して複数のディスクＤ００～Ｄ０Ｎに動作のための電力を供給し、また、電力線１０７及び接続インターフェース１３０、及び電力線１０４を介してディスクアレイコントローラ１１，１２の動作のための電力を供給する。同様に、電源１０６は、電力線１０７及び接続インターフェース１３１、１３２を介し、複数のディスク装置Ｄ１０～１Ｎ、Ｄ２０～Ｄ２Ｎに動作のための電力を供給する。
【００２１】
各ディスク装置Ｄ００～Ｄ２Ｎは、ハードディスクドライブであり、例えば、ＡＴＡ規格のハードディスクドライブやＳＡＳ規格のハードディスクドライブが用いられる。
【００２２】
管理用端末３１は、ディスクアレイ装置１に対する保守管理を実行するために用いられる端末装置である。管理用端末装置３１には、管理画面３２が備えられており、管理者は管理画面３２を通じて、ディスクアレイ装置１の状態を管理する。
【００２３】
図３を参照して、ディスクアレイコントローラ１１の内部構成について説明する。なお、ディスクアレイコントローラ１２に関しても同様の内部構成を有している。ディスクアレイコントローラ１１は、マルチコアＣＰＵ１１０、不揮発性メモリ１１１、揮発メモリ１１２、ホスト側物理ポート１１３、ドライブ側物理ポート１１４、管理ネットワーク用物理ポート１１５、ブートＲＯＭ１１６を備えている。
【００２４】
不揮発性メモリ１１１には、ディスクアレイ装置１を制御するための制御プログラムが格納されている。不揮発性メモリ１１１は、電力の供給が停止しても継続的にデータを記憶しておくことができ、例えば、Ｆｌａｓｈメモリが用いられる。制御プログラム１１９については、図６を用いて後述する。
【００２５】
揮発性メモリ１１２は、ディスク装置Ｄから読み出したデータ、ディスク装置Ｄに書き込むデータ、及び、マルチコアＣＰＵ１１０による演算結果を一時的に格納するためのデータバッファ領域１１２と、マルチコア１１０によって制御プログラム１１９を実行するために、制御プログラム１１９を読み出して格納しておく制御プログラム配置領域１１７とを備える。揮発性メモリは停電などによりメモリへの電力供給が停止するとデータを記憶しておくことができなくなるメモリであり、例えばＤＲＡＭ（Ｄｙｎａｍｉｃ Ｒａｎｄｕｍ Ａｃｃｅｓｓ Ｍｅｍｏｒｙ）などが用いられる。
【００２６】
ホスト側物理ポート１１３は、ストレージネットワーク４０へ物理的に接続し、ホスト２０～２２と電気的信号を送受信するための伝送路の受け口である。
【００２７】
ドライブ側物理ポート１１４は、接続インターフェースと電気的信号を送受信するための伝送路の受け口である。
【００２８】
管理ネットワーク用物理ポート１１５は、管理用ネットワークへ接続し、管理用端末装置３１と電気的信号を送受信するための受け口である。
【００２９】
ブートＲＯＭ１１６は、ディスクアレイ装置１を起動した際、不揮発性メモリ１１１に格納された制御プログラム１１９を、揮発性メモリ１１２の制御プログラム配置領域１１

７に読み出すためのイニシャルプログラムローダを格納した読み出し専用メモリ（Ｒｅａｄ
Ｏｎｌｙ Ｍｅｍｏｒｙ）である。

【００３０】

マルチコアＣＰＵ１１０は、複数のプロセッサコア（ＭＰＵコア）を内蔵する演算処理装置であり、信号線１２０を介して、不揮発性メモリ１１１、揮発性メモリ１１２、ホスト側物理ポート１１３、ドライブ側物理ポート１１４、管理ネットワーク用物理ポート１１５、ブートＲＯＭ１１６と相互に接続されている。マルチコアＣＰＵ１１０は、信号線１１６を介して、不揮発性メモリ１１１からのデータの読み出しと書き込み、揮発性メモリ１１２からのデータの読み出しと書き込みを実行するほか、信号線１２０とホスト側物理ポート１１３を介してホスト２０～２２との間でコマンドおよびデータの送受信を実行し、信号線１２０とドライブ側物理ポート１１４、及び、図１に示した信号線１０２、１０３、接続インターフェース１３０～１３２を介して、ディスク装置Ｄとの間でコマンドおよびデータの送受信を実行する。

【００３１】

また、マルチコアＣＰＵ１１０は、信号線１０１を介して他系コントローラと相互に接続されているおり、他系コントローラとの間でデータおよびコマンドの送受信を実行する。図３はディスクアレイコントローラ１１を示しているので、他系コントローラとは、具体的には、ディスクアレイコントローラ１２を指す。なお、ディスクアレイコントローラ１２に対する他系コントローラとはディスクアレイコントローラ１１を意味する。

【００３２】

図４は、ディスクアレイコントローラ１１の概観図を示したものである。なお、ディスクアレイコントローラ１２の概観図も同様である。

【００３３】

ディスクアレイコントローラ１１には、マルチコアＣＰＵ１１０をはじめとし、図３の内部構成で示した部品が基板回路上に配置され、接続されているほか、接続コネクタ１２２、ブラケット１２１、ホスト接続用ポート１２３、管理ネットワーク接続用ポート１２４、エラー表示ＬＥＤ１２５が備えられている。

【００３４】

また、図４では記載を省略するが、マルチコアＣＰＵ１１０や周辺回路上の半導体の上には、発熱による破壊を防ぐ為の放熱板やファンが取り付けられることもある。

【００３５】

接続コネクタ１２２は接続インターフェース１３０上に設けられたコネクタと嵌合することによって、信号線１０１、１０２、及び電力線１０４と物理的な接続を行う。接続コネクタ上にはこのほかにディスクアレイ装置上で必要となるいくつかの信号線を備えている。たとえば、ディスクアレイコントローラ挿抜検出用信号線などである。

【００３６】

ホスト接続用ポート１２３はストレージネットワーク４０へ接続するためのケーブルのコネクタを挿入する接続口であり、管理ネットワーク接続用ポート１２４は、管理用ネットワーク３０へ接続するためのケーブルのコネクタを挿入する接続口である。

【００３７】

エラー表示ＬＥＤ１２５はディスクアレイコントローラ１１に障害が発生し保守交換が必要となった際に、エラーが発生していることを視覚的にディスクアレイ装置１の外部に通知するための表示灯である。

【００３８】

図５を参照して、ディスクアレイコントローラ１１に搭載されているマルチコアＣＰＵ１１０の機能的構成の一例について説明する。マルチコアＣＰＵ１１０は、複数のＭＰＵコア（プロセッサコア）１１１０～１１１Ｎと、内部スイッチ１１２０、不揮発メモリコントローラ１１３０、ディスクアレイコントローラ間転送コントローラ１１４０、パリティ演算器１１５０、ＣＰＵ内部キャッシュ１１６０、揮発メモリコントローラ１１７０を

備える。複数のＭＰＵコア１１１０～１１１Ｎはそれぞれした独立したプロセッサであるが、1つの半導体素子上に作成されており物理的に個別に切り離して利用することはできない点が、従来のマルチプロセッサとは異なる。

【００３９】

内部スイッチ１１２０は複数のＭＰＵコア１１１０～１１１Ｎ、不揮発メモリコントローラ１１３０、ディスクアレイコントローラ間転送コントローラ１１４０、パリティ演算器１１５０、ＣＰＵ内部キャッシュ１１６０、揮発メモリコントローラをスイッチ機構によって高速に相互接続している。

【００４０】

不揮発性メモリコントローラ１１３０はＭＰＵコア１１１０～１１１Ｎの指示に基づいて不揮発性メモリ１１１との間でデータの転送を実行するＩ／Ｏコントローラである。

【００４１】

ディスクアレイコントローラ間転送コントローラ１１４０は、他系のディスクアレイコントローラに搭載されたマルチコアＣＰＵ１１０とコマンドとデータの送受信を行う。

【００４２】

パリティ演算器１１５０は、各ＭＰＵコア１１１０～１１１Ｎの指示に基づいて与えられたデータのパリティ生成や整合性の確認を高速に行うのに用いられる。たとえばＸＯＲ演算器が含まれる。

【００４３】

ＣＰＵ内部キャッシュ１１６０は、複数のＭＰＵコア１１１０～１１１Ｎからマルチコアプロセッサ１１０外部にある揮発性メモリ１１２よりも高速にアクセスできる揮発性のメモリである。ＣＰＵ内部キャッシュ１１６０は複数のＭＰＵコア１１１０～１１１Ｎが演算結果を一時的に格納するほか、ＭＰＵコアの動作状況を格納するＭＰＵコア管理テーブルなどを格納するのに用いられる。

【００４４】

揮発性メモリコントローラ１１７０はＭＰＵコア１１１０～１１１Ｎの指示に基づいて揮発性メモリ１１２との間でデータの転送を実行するＩ／Ｏコントローラである。

【００４５】

以上の様に、本発明のマルチコアＣＰＵ１１０は複数のＭＰＵコアのほかにＩ／Ｏコントローラなどの周辺制御回路を内蔵しているが、少なくとも複数のＭＰＵコアを内蔵していれば他の構成を取っていても良い。例えば不揮発メモリコントローラは内蔵されずに外部回路として設けられる場合や、マルチコアＣＰＵにＴＣＰ／ＩＰコントローラが内蔵されるような場合である。

【００４６】

図６を参照して、制御プログラム１１９の詳細について説明する。制御プログラム１１９は、監視制御プログラムモジュールＰｒ１，運用管理プログラムモジュールＰｒ２、ＲＡＩＤ制御プログラムモジュールＰｒ３，ＮＡＳ制御プログラムモジュールＰｒ４、ホストＩ／Ｏ制御プログラムモジュールＰｒ５、ドライブＩ／Ｏ制御プログラムモジュールＰｒ６、初期ＭＰＵコア割当管理テーブルＴｂ１、初期閾値管理テーブルＴｂ２，ＲＡＩＤ設定管理テーブルＴｂ３，ＬＵ設定管理テーブルＴｂ４、ＮＡＳボリューム管理テーブルＴｂ５を備えている。

【００４７】

監視制御プログラムモジュールＰｒ１は、プログラムモジュールＰｒ２～Ｐｒ６の動作管理及びＭＰＵコアの資源管理を行う、特権レベルの最も高いプログラムモジュールであり、他のプログラムモジュールの処理負荷状況に応じて自律的にＭＰＵコアの割当て数の変更を実行する。

【００４８】

運用管理プログラムモジュールＰｒ２は、監視制御プログラムモジュールＰｒ１に次いで特権の高いプログラムモジュールであり、管理用端末装置３１からのディスクアレイ装置１に対する運用・保守管理に関する設定を受付け、他のプログラムモジュールを通じて

実際に設定を実行するプログラムモジュールである。運用・保守管理には、例えば、ＲＡＩＤグループの設定、ＬＵ（Ｌｏｇｉｃａｌ　Ｕｎｉｔ）の設定、ＮＡＳ（Ｎｅｔｗｏｒｋ　Ａｔｔａｃｈｅｄ　Ｓｔｏｒａｇｅ）の設定、ホストマッピングの設定、監視機能の設定、制御プログラムのアップデート、稼働状況の確認、動作ログの閲覧・取得、ディスク表面検査の設定、ディスクアレイ装置の起動・停止などがある。

【００４９】

ＲＡＩＤ制御プログラムモジュールＰｒ３は、運用管理プログラムモジュールＰｒ２を介して、管理用端末装置３１から管理用端末画面３２を通じた指定内容に基づき、ＲＡＩＤグループの作成やＬＵの作成、ＬＵの初期化、ホスト２０～２２に対するＬＵとＬＵ番号（ＬＵＮ）との対応付けなどを行うほか、ホスト２０～２２から受領したコマンドの解釈を行い、必要であれば演算を行いながら、適切なディスク装置Ｄに対しコマンドを発行してデータの読み書きを実行し、ホスト２０～２２との間でコマンド処理の結果通知やデータの送受信を実行する。ＲＡＩＤ制御プログラムはＰｒ３は、また、次に説明するＮＡＳ制御プログラムとの間でも同様のコマンド送受信、演算、ディスク装置Ｄへのアクセスを実行する。

【００５０】

ＮＡＳ制御プログラムモジュールＰｒ４は、ホスト２０～２２からのファイルレベルアクセス処理を実行するプログラムモジュールであり、その他にＲＡＩＤ制御プログラムモジュールＰｒ３によって作成されたＬＵ上へのファイルシステムの構築や構築されたファイルシステムサイズの変更、ファイルやディレクトリへのアクセス権の設定、アクセス認証方法の設定などを行う。

【００５１】

ホストＩ／Ｏ制御プログラムモジュールＰｒ５は、ＲＡＩＤ制御プログラムモジュールＰｒ３が作成した転送リストの指示に基づき、ホスト２０～２２と揮発性メモリ１１２にあるデータバッファ領域１１８との間で、パリティ演算器１１５０を介してデータ整合性のチェックを行いながら、データ転送を実行するプログラムモジュールである。ドライブＩ／Ｏ制御プログラムモジュールＰｒ６は、同様にＲＡＩＤ制御プログラムモジュールＰｒ４が指定する転送リストに基づき、ディスク装置Ｄと揮発性メモリ１１２にあるデータバッファ領域１１８との間で、パリティ演算器１１５０を介してデータ整合性のチェックを行いながら、データ転送を実行するプログラムモジュールである。

【００５２】

初期ＭＰＵコア割当管理テーブルＴｂ１は、ディスクアレイ装置１が起動した際の、ＭＰＵコア資源管理内容と、各制御プログラムモジュールＰｒ１～Ｐｒ６の初期割当を指定した管理表である。初期ＭＰＵコア割当て管理テーブルＴｂ１の詳細は、図７を用いて後述する。

【００５３】

初期閾値管理テーブルＴｂ２は、各プログラムモジュールＰｒ１～Ｐｒ６の負荷の上限閾値の初期値を管理しているテーブルである。初期閾値管理テーブルＴｂ２の詳細は、図８を用いて後述する。

【００５４】

ＲＡＩＤ設定管理テーブルＴｂ３は、ＲＡＩＤグループを構成するディスク装置についての各種情報を管理するために用いられるテーブルであり、例えば、ＲＡＩＤグループＮｏ、ＲＡＩＤグループの総記憶容量，ＲＡＩＤレベル、ＲＡＩＤグループを構成するディスク装置、正常・異常の状態などの情報を管理する。

【００５５】

ＬＵ設定管理テーブルＴｂ４は、ＲＡＩＤグループ上に作成される論理ユニット（Ｌｏｇｉｃａｌ　Ｕｎｉｔ）を管理するためのテーブルであり、例えば、ＬＵＮ、所属するＲＡＩＤグループＮｏ、論理ユニットに設定されている記憶容量、正常・異常の状態などの情報を管理する。

【００５６】

ＮＡＳボリューム設定管理テーブルＴｂ５は、ＮＡＳが作成するファイルシステムを管理するテーブルであり、例えば、ファイルシステム名、ファイルシステムを構成するＬＵの番号、ファイルシステムのフォーマット、ファイルシステムの総容量、ファイルシステムの使用容量、ファイルシステムの差分管理の有無、差分管理容量、差分管理容量の使用率、ファイルシステムの状態などの情報を管理する。

【００５７】

図７に初期ＭＰＵコア割当管理テーブルＴｂ１の一例を示す。図７に示す初期ＭＰＵコア割当管理テーブルＴｂ１は、管理上振付けたＭＰＵコア番号、同ＭＰＵを初期の段階で使用するプログラムモジュール名、モジュールの割当形式、特権レベル、グループＮｏについての情報を保持している。

【００５８】

モジュールの割当形式は、固定・半固定・動的の３種類がある。固定は初期ＭＰＵコア割当管理テーブルで指定したＭＰＵコア以外を割当てることができないことを示し、半固定は初期ＭＰＵコア割当管理テーブルで指定したＭＰＵコアに加え、他のＭＰＵコアを負荷分散用に割当可能であることを示す。動的は具体的なプログラムモジュールが初期の段階では割当てられず、未使用で開始することを示している。図７を例に取ると、監視制御プログラムモジュールと運用管理プログラムモジュールはそれぞれＭＰＵコア０、ＭＰＵコア１に固定的に割当てられ変更不可であり、ＲＡＩＤ制御は当初ＭＰＵコア２が割当てられるが、必要に応じ他のコアも割当可能であることを示している。

【００５９】

特権レベルは、プログラムモジュール間での制御、非制御の関係を表す整数値であり、小さいほど特権性が高いことを示す。図７を例に取ると、監視制御プログラムモジュールは最も高い特権レベル０であり、他のいずれのプログラムモジュールからも制御指示を受けず、逆にいずれのプログラムモジュールにも制御指示が可能であることが分かる。運用管理モジュールの特権レベルは１であり、これより高い監視制御プログラムモジュールからの制御指示は受けるが、それ以外からは制御指示を受けない。逆に、例えば、特権レベルが１つ低い、特権レベル２のＲＡＩＤ制御プログラムモジュールに対しては制御指示が可能である。なお、特権レベルが同じとは、お互いに独立している場合などであり、制御指示をお互いに与えることはない。

【００６０】

グループＮｏは、負荷分散で融通できるＭＰＵコアのグループを示している。図７を例に取れば、ＭＰＵコアＮｏ６及び７のＭＰＵコアは、グループＮｏが００３なので、同じグループＮｏを持つＭＰＵコアＮｏ２、３に割当てられているプログラムモジュールの負荷分散用に利用が可能である。

【００６１】

図８は、初期閾値管理テーブルＴｂ２の一例を示したものである。図８に示す初期閾値管理テーブルＴｂ２は、使用モジュールと、負荷分散要と判断するための負荷閾値、負荷分散開始を決定する閾値オーバ回数についての情報を保持している。図８を例に取ると、監視制御プログラムモジュールは負荷閾値に１００が指定されており、負荷は１００％を超えることはないから、負荷が高くなっても分散しないことを示している。一方、ＲＡＩＤ制御プログラムモジュールの負荷閾値は９０％であり、割当ＭＰＵコアの負荷が９０％を超えると閾値オーバ回数１回としてカウントされる。負荷は定期的に監視され、閾値オーバ回数が閾値オーバ規定回数欄に記載されている５回を超えると負荷分散を開始する。

【００６２】

図９はＣＰＵ内部キャッシュ内で監視制御プログラムモジュールが管理するＭＰＵコア割当テーブルの一例を示す図である。図９に示すＭＰＵコア割当テーブルには、ＭＰＵコアＮｏ、ＭＰＵコアの使用負荷率、ＭＰＵコアのグループＮｏ、状態、ＭＰＵコアを使用しているプログラムモジュール名、モジュールの割当形式、特権レベル、及び、負荷閾値のオーバ回数について情報を保持している。

【００６３】

図１０は管理用端末３１の管理端末画面３２においてＭＰＵコア利用状況の情報を確認した際の画面を示している。画面にはＭＰＵコアＮｏとそのＭＰＵコアで動作しているプログラムモジュール名、状態、負荷状況が表示され、情報が更新されるたびに書き換えられる。

【００６４】

図１１～１３を参照して本実施例に係るディスクアレイ装置の制御方法について説明する。

【００６５】

図１１を参照して、本実施例に係るディスクアレイ装置１の装置起動時において制御プログラムをマルチコアＣＰＵの各ＭＰＵコアに割当てる処理手順について説明する。図１１は装置起動時の制御プログラムのロードとマルチコアＭＰＵ１１０の各ＭＰＵコア１１０～１１１Ｎへのプログラムモジュール割当処理の実行手順を示すフローチャートである。

【００６６】

図１１に示すフローチャートは、ディスクアレイ装置１の電源スイッチがＯＮにされ、ディスクアレイコントローラ１１、１２への通電が開始された際に実行される。

【００６７】

ディスクアレイコントローラ１１、１２への通電が開始されると、マルチコアＣＰＵ１１０は、ブートＲＯＭ上にあるイニシャルプログラムローダを実行する（ステップＳ１０００）。イニシャルプログラムローダは、マルチコアＣＰＵ１１０の動作チェックと揮発性メモリ１１２のチェックを実施する（ステップＳ１０１０）。マルチコアＣＰＵの動作チェックとは、一例としては、各ＭＰＵコア１１１０～１１１Ｎが正常であるか否か確認させるＣＰＵのセルフチェック機能が挙げられる。また揮発性メモリのチェックとはメモリへ特定パターンの値をライトした後、正しくリードできるか否か確認するチェック方法がある。

【００６８】

イニシャルプログラムローダによってチェックした結果が正常であった場合には（ステップＳ１０２０：Ｙｅｓ）、次の処理に進み、制御プログラム１１９を揮発性メモリ１１２の制御プログラム配置領域１１７にロードし、イニシャルプログラムローダから制御プログラムへと処理を移す（ステップＳ１０３０）。

【００６９】

制御プログラムモジュールでは、最初に監視制御プログラムモジュールＰｒ１が起動される。そしてＣＰＵ内部キャッシュ１１６０上にＭＰＵコア管理テーブルＴｂ６を作成し、監視制御プログラムモジュールＰｒ１の情報を登録する（ステップＳ１０４０）。その後、監視制御プログラムモジュールＰｒ１は初期ＭＰＵコア割当管理テーブルＴｂ１を参照し、他のモジュールに使用ＭＰＵコアを割り当て、ＭＰＵコア管理テーブルＴｂ６の情報を更新する（ステップＳ１０５０）。次に、初期閾値管理テーブルＴｂ２に基づき、負荷閾値と閾値オーバ回数規定値を設定する（ステップＳ１０６０）。ＭＰＵコアを割り当てられた各プログラムモジュールは、起動して各制御処理を開始する（ステップＳ１０７０）。

【００７０】

これらの一連の作業は、管理用端末３１およびその管理画面３２を通じて、管理者に報告される（ステップＳ１０８０）。

【００７１】

一方、起動後にイニシャルプログラムロードがマルチコアＣＰＵ，揮発性メモリをチェックした結果が正常でなかった場合には（ステップＳ１０２０：Ｎｏ）、本ディスクアレイコントローラは動作不可能であると判断し、エラーを通知する。エラー通知とは例えば、イニシャルプログラムローダによる管理用端末３１への通知や装置に搭載されたスピーカーによるアラーム音などの発生による。

【００７２】

図１２を参照して、本実施例に係るディスクアレイ装置１の制御プログラム内監視制御プログラムモジュールＰｂ１による自律負荷分散処理について説明する。図１２は、監視制御プログラムモジュールＰｂ１が過負荷を検出し、当該過負荷プログラムモジュールの処理を負荷分散させるために追加のＭＰＵコアを割り当てる処理手順を示すフローチャートである。

【００７３】

図１２のフローチャートは、ディスクアレイ装置１の電源スイッチがＯＮにされ、ディスクアレイコントローラ１１、１２を含む基本筐体、および、増設筐体に通電が開始され、図１１に示した手順によって制御プログラムの起動処理が完了した後に、装置が停止されるまで監視制御プログラムモジュールによって繰り返し実行される。

【００７４】

監視制御プログラムモジュールは、動作中の各プログラムモジュールへの使用しているＭＰＵコアごとの使用率（負荷状態）の問い合わせ情報を取得するとともに、ＭＰＵコア管理テーブルＴｂ６の情報を更新する（ステップＳ２０００）。次に、監視制御プログラムモジュールＰｒ１は、初期閾値管理テーブルＴｂ２に基づいて設定された負荷閾値を照らし合わせ、各負荷状態が閾値を超えているモジュールがないかチェックする。もし負荷閾値を超えているモジュールがあれば（ステップＳ２０１０：Ｙｅｓ）、閾値を超えている当該ＭＰＵコアの閾値オーバ回数を１増加させ、ＭＰＵコア管理テーブルＴｂ６上に反映する。次に、当該閾値オーバ回数が、規定回数を超えていないかチェックする。閾値オーバ回数が規定回数Ｎを越える値となった場合（ステップＳ２０３０：Ｙｅｓ）、監視制御プログラムモジュールＰｒ１は、当該プログラムモジュール処理の過負荷状態が続いており、負荷分散が必要であると判断する。監視制御プログラムモジュールＰｒ１は、ＭＰＵコア管理テーブルＴｂ６を参照し、当該過負荷プログラムモジュールが割り当てられているＭＰＵコアのグループＮｏを取得後、同一グループＮｏを持つ未割当のＭＰＵコアを検索する（ステップＳ２０４０）。未割当ＭＰＵコアが存在した場合（ステップＳ２０５０：Ｙｅｓ）は、当該過負荷モジュールに未割当ＭＰＵコアを割当て、ＭＰＵ管理テーブルＴｂ６の情報を更新し、当該未割当コア欄に動作プログラムモジュール名の登録と、特権レベルの変更を行う。また、過負荷判定されたＭＰＵコアの閾値オーバ回数をゼロにクリアする。そして、これらの変化状況を管理端末に通知して（ステップＳ２２３０）処理を終了する。

【００７５】

一方、当該過負荷モジュールに現在割り当てられているＭＰＵコアと同一のグループＮｏを持つ未割当ＭＰＵコアが存在しない場合（ステップＳ２０５０：Ｎｏ）は、グループＮｏが－１、すなわちどのグループにも属していない未割当ＭＰＵコアを検索する（ステップＳ２０６０）。

【００７６】

どのグループにも属していない未割当ＭＰＵコアが存在した場合（ステップＳ２０７０：Ｙｅｓ）は、当該過負荷モジュールに未割当ＭＰＵコアを割当て、ＭＰＵ管理テーブルＴｂ６の情報を更新し、当該未割当コア欄に動作プログラムモジュール名の登録と、特権レベルの変更を行う。また、過負荷判定されたＭＰＵコアの閾値オーバ回数をゼロにクリアする。そして、これらの変化状況を管理端末に通知して（ステップＳ２２３０）処理を終了する。

【００７７】

どのグループにも属していない未割当ＭＰＵコアが存在しなかった場合（ステップＳ２０７０：Ｙｅｓ）は、未割当ＭＰＵコアの利用はできないので、すでに利用されている動的割当のＭＰＵコアのうち、もっとも利用率の低いＭＰＵコアの割当を変えることを考え、当該過負荷モジュールに現在割り当てられているＭＰＵコアと同一のグループＮｏを持ち、割当方式が「動的」であるＭＰＵコアを検索する（ステップ２０９０）。検索して見つかった動的割当のＭＰＵコアの負荷状態が、当該過負荷モジュールの使用しているＭＰＵコアの負荷値の２倍よりも大きい場合（ステップＳ２０９０：Ｙｅｓ）、すなわち、過

負荷プログラムモジュールの処理を２つのＭＰＵコアに負荷分散した際の１つあたりのＭＰＵコア負荷が現在割り当てられている処理の負荷よりも大きい場合、現在割り当てられている処理を終了し、当該動的割当ＭＰＵコアの特権レベルを２５５に変更し、未割当状態にする。（ステップＳ２２１０）。その後、当該過負荷モジュールに未割当ＭＰＵコアを割当て、ＭＰＵ管理テーブルＴｂ６の情報を更新し、当該未割当コア欄に動作プログラムモジュール名の登録と、特権レベルの変更を行う。また、過負荷判定されたＭＰＵコアの閾値オーバ回数をゼロにクリアする。そして、これらの変化状況を管理端末に通知して（ステップＳ２２３０）処理を終了する。

【００７８】

検索して見つかった動的割当のＭＰＵコアの負荷状態が、当該過負荷モジュールの使用しているＭＰＵコアの負荷値の２倍と同等かそれよりも大きい場合（ステップＳ２０９０：Ｎｏ）、当該プログラムモジュールの負荷分散を断念し、管理端末３１に高負荷状態であることを通知し、処理を終了する。

【００７９】

なお、図１２の説明の冒頭に述べた、各ＭＰＵコアの負荷状態監視の際に、負荷閾値を超えたモジュールが存在しない場合（ステップＳ２０１０：Ｎｏ）や、存在しても閾値オーバ回数が規定回数未満（ステップ２０３０：Ｎｏ）だった場合には、負荷分散のためのＭＰＵ割当変更処理は不要であると判断し、一定時間待ち（ステップＳ２２００）、間隔を置いた後、負荷状態の監視を繰り返す。

【００８０】

図１３を参照して、本実施例に係るディスクアレイ装置のディスクアレイコントローラにおいて、マルチコアＣＰＵの一部ＭＰＵコアに障害が発生した際の処理について説明する。マルチコアＣＰＵの内部には、複数のＭＰＵコアが存在するが、物理的に切り離すことができないため、あるＭＰＵコアに物理障害が生じた場合には、ＣＰＵ全体を交換する必要がある。図１３は、この障害対応のため、動作中のプログラムモジュールをフェイルオーバさせるために停止する処理手順を示したフローチャートである。

【００８１】

図１３に示すフローチャートは、マルチコアＣＰＵ１１０のうち、ある特定のＭＰＵコア１１１ｘ（ｘは整数）に障害が発生した際に実行される。

【００８２】

監視制御プログラムＰｒ１は、あるＭＰＵコア１１１ｘに障害が発生したのを検出すると（ステップ３０００）、ＭＰＵコア割当テーブルＴｂ６を参照してＭＰＵコア１１１ｘの割当状況を確認するとともに、状態を「障害」に変更する（ステップＳ３０１０）。具体的には、割当済みであるか否かと、割当済みである場合の割当形式について確認する。

【００８３】

障害を発生したＭＰＵコア１１１ｘが固定割当で利用されている場合（ステップＳ３０２０：Ｙｅｓ）、ＭＰＵコア１１１ｘは他の処理の監視制御や管理用端末３１による運用管理制御を行っているコアであるため、正常なフェイルオーバ処理が行えなくなる可能性がある。そこで、監視制御プログラムＰｒ１は動作可能であれば、管理端末３１への障害通知と保守交換用のエラー表示ＬＥＤ１２５の点灯を試み（ステップＳ３２００）、信号線１０１を通じて他系のディスクアレイコントローラに障害情報を送信して、処理の引継ぎ依頼を通知する（ステップＳ３０７０）。なお、ステップＳ３２００、Ｓ３０７０に示す処理が行えないような重大な障害が発生している場合には、信号線１０１を介したディスクアレイコントローラ間の定期的な通信が断絶するため、他系のディスクアレイコントローラによってこの障害を検出される。

【００８４】

他系のディスクアレイコントローラは、ステップＳ３０７０に示すように処理引継ぎの依頼を受けた場合や、もしくはディスクコントローラ間にある信号線１０１を介した通信できなくなった場合には、もう一方のディスクアレイコントローラに障害が発生したものと判断し、信号線を介して、障害が発生しているディスクアレイコントローラのマルチコ

アＣＰＵ１１０にリセット信号を常時印加する。常時リセット信号が印加されていることにより、障害マルチコアＣＰＵ１１０はリセットされ続けるので、それ以上動作することができなくなる。このため、ＭＰＵコア１１１ｘの誤動作によってデータを破壊してしまうような恐れがなくなる。なお、ここでは常時リセットにより動作を停止させているが、給電の停止など他の手段によってもよい。

【００８５】

障害を発生したＭＰＵコア１１１ｘが半固定割り当てで利用されている場合、監視制御プログラムモジュールＰｒ１は、ＭＰＵコア割当テーブルＴｂ６を確認し、障害を発生したＭＰＵコア１１１ｘと同一のプログラムモジュールが存在しないか検索する（ステップＳ３３００）。

【００８６】

同一のプログラムモジュールが見つかった場合には、監視制御プログラムモジュールＰｒ１は当該プログラムが動作しているＭＰＵコアの割当形式を動的から半固定に変更し（ステップＳ３３１０）、その後管理端末に障害を通知して、保守交換用のＬＥＤを点灯させる（ステップＳ３０４０）。

【００８７】

次に監視制御プログラムは、ＭＰＵコア割当テーブルＴｂ６を参照して、特権レベルの低いプログラムモジュールから順に終了していく（ステップ３０５０）。

【００８８】

以降ステップＳ３０７０～Ｓ３０８０に処理を進めるが、ステップＳ３０７０～Ｓ３０８０については、既に説明しているため省略する。

【００８９】

一方、障害コアに割り当てられていたプログラムモジュールと同一のモジュールが割当られたＭＰＵコアが存在しない場合には、ステップＳ３０４０に処理を進める。

【００９０】

障害を発生したＭＰＵコア１１１が、固定割当でもなく（ステップＳ３０２０：Ｎｏ）、また半固定でもない場合（ステップＳ３０３０：Ｎｏ）、すなわち動的で割り当てられている場合には、監視制御プログラムモジュールＰｒ１はステップＳ３０４０～Ｓ３０８０に示す処理を行う。なお、ステップＳ３０４０～Ｓ３０８０の処理については先に説明しているため省略する。

【００９１】

以上の実施例によれば、本発明にかかるディスクアレイ装置は、複数個あるＭＰＵコアを管理テーブルを用いて固定的または半固定的または動的な資源として管理するために、他の障害ですべての処理に影響が生じるような問題を回避することができる。また各ＭＰＵコアの負荷状況を監視することで、ＣＰＵ資源をコア単位で効率よく活用することができる。

【００９２】

また、障害発生時に特権レベルの確認によって動作中のプログラムモジュールの依存関係を考慮しつつフェイルオーバすることが可能であり、マルチコアＣＰＵで生じる特定コアの障害時の対応を行うことが可能である。

【００９３】

以上実施例で説明したが、本発明の他の実施形態１は、前記一の単位プロセッサは、非動的になるように割り当てる制御プログラムのうちの１以上を、非動的になるように割り当てる単位プロセッサとは別の単位プロセッサに動的になるように割り当てるディスクアレイ装置である。

【００９４】

本発明の他の実施形態２は、前記一の単位プロセッサは、非動的になるように割り当てる制御プログラムのうちの１以上を、非動的になるように割り当てる単位プロセッサのみに割り当てるディスクアレイ装置である。

【００９５】

本発明の他の実施形態３は、前記一の単位プロセッサは、非動的になるように制御プログラムを割り当てた単位プロセッサの動作負荷が相対的に高いとき、動的になるように割り当てた制御プログラムの動作負荷が相対的に低い単位プロセッサの処理を終了させ、終了させた単位プロセッサに、前記相対的に動作負荷が高い単位プロセッサに割り当てた制御プログラムを、動的になるように割り当てるディスクアレイ装置である。

【００９６】

本発明の他の実施形態４は、前記一の単位プロセッサは、制御プログラムの動作負荷が相対的に高い単位プロセッサが属するグループ内で、かつ、動的になるように割り当てた制御プログラムの動作負荷が相対的に低い単位プロセッサの処理を終了させ、終了させ単位プロセッサに、前記相対的に動作負荷が高い単位プロセッサに割り当てた制御プログラムを、動的になるように割り当てるディスクアレイ装置である。

【００９７】

本発明の他の実施形態５は、前記一の単位プロセッサは、どのグループにも属さない単位プロセッサに、前記相対的に動作負荷が高い単位プロセッサに割り当てた制御プログラムを、動的になるよう割り当てるとともに、割り当てた単位プロセッサを、前記相対的に動作負荷が高い単位プロセッサが属するグループに属するとするディスクアレイ装置である。

【００９８】

本発明の他の実施形態６は、前記ディスクアレイコントローラは、物理的に分離不可能な複数個のプロセッサコアを内蔵したＣＰＵを搭載しており、各プロセッサコアはそれぞれ単位プロセッサとなり、一の単位プロセッサは、すべての単位プロセッサを個別に管理し、各々完結した制御プログラムを、すべての単位プロセッサにプロセッサ単位で割当てる際に、適宜動作終了可能とする動的になるように、又は前記ＣＰＵ全体の動作終了まで動作可能とする非動的になるように割り当て、割り当てた制御プログラムの動作負荷又は動作状態を単位プロセッサごとに管理し、特定のプロセッサコアに障害が発生した場合に、該プロセッサコアの単位プロセッサに制御プログラムを動的に又は非動的になるように割り当てたか及び制御プログラムの終了順を考慮し、すべての制御プログラムを終了させるディスクアレイ装置である。

【００９９】

本発明の他の実施形態７は、前記一の単位プロセッサは、前記障害が発生したプロセッサコアの単位プロセッサに割り当てた制御プログラムが非動的になるよう割り当てたものであり、かつ、前記制御プログラムを動的になるよう割り当てた別の単位プロセッサがあるとき、前記制御プログラムを、前記別の単位プロセッサに非動的になるよう割り当てたとするディスクアレイ装置である。

【０１００】

本発明の他の実施形態８は、前記一の単位プロセッサは、前記障害が発生したプロセッサコアの単位プロセッサに割り当てた制御プログラムが該単位プロセッサのみに非動的になるように割り当てた制御プログラムであるとき、前記ＣＰＵ全体の動作を終了させるディスクアレイ装置である。

【０１０１】

本発明の他の実施形態９は、前記ディスクアレイコントローラは、物理的に分離不可能な複数個のプロセッサコアを内蔵したＣＰＵを搭載しており、内部スイッチ、不揮発性メモリコントローラ、揮発性メモリコントローラ、ディスクアレイコントローラ間転送コントローラ、パリティ演算器及びＣＰＵ内部キャッシュを有し、前記制御プログラムは、監視制御プログラムモジュール、運用管理プログラムモジュール、ＲＡＩＤ制御プログラムモジュール、ＮＡＳ制御プログラムモジュール、ホストＩ／Ｏ制御プログラムモジュール、ドライブＩ／Ｏ制御プログラムモジュール、初期ＭＰＵコア割当管理テーブル、初期閾値管理テーブル、ＲＡＩＤ設定管理テーブル、ＬＵ設定管理テーブル、及びＮＡＳボリューム管理テーブルを備えており、前記一の単位プロセッサは、自己及び他の単位プロセッサの制御プログラムの負荷状態を監視し、負荷閾値及び閾値オーバ回数が規定回数を超え

ると、同一グループ内の未割当コアを検索し、検出した未割当コアに、前記制御プログラムを割り当てるディスクアレイ装置である。

【０１０２】

本発明の他の実施形態１０は、物理的に分離不可能な複数個のプロセッサコアを内蔵したＣＰＵを搭載し、各プロセッサコアはそれぞれ単位プロセッサとなるディスクアレイコントローラを備えるディスクアレイ装置の制御方法であって、すべての単位プロセッサを個別に管理し、各々完結した制御プログラムを、すべての単位プロセッサにプロセッサ単位で割当てる際に、適宜動作終了可能とする動的になるように、又は前記ＣＰＵ全体の動作終了まで動作可能とする非動的になるように割り当て、割り当てた制御プログラムの動作負荷又は動作状態を単位プロセッサごとに管理するディスクアレイ装置の制御方法である。

【図面の簡単な説明】

【０１０３】

【図１】本実施例に係るディスクアレイ装置の概略構成を示す説明図。

【図２】本実施例に係るディスクアレイ装置の概観図。

【図３】本実施例に係るディスクアレイ装置におけるディスクアレイコントローラの機能的構成を示すブロック図。

【図４】本実施例に係るディスクアレイ装置におけるディスクアレイコントローラの概観図。

【図５】ディスクアレイコントローラに搭載される複数プロセッサコア（ＭＰＵコア）を内蔵したＣＰＵ（マルチコアＣＰＵ）の機能的構成の一例を示すブロック図。

【図６】制御プログラムに含まれる各プログラムモジュール及び管理テーブルを示す説明図。

【図７】制御プログラムの初期ＭＰＵコア割当管理テーブルの一例を示す図。

【図８】制御プログラムの初期閾値管理テーブルの一例を示す図。

【図９】ＭＰＵコア割当テーブルの一例を示す図。

【図１０】管理用端末装置３１の管理画面３２におけるＭＰＵコア利用状況に関する表示画面。

【図１１】制御プログラムをマルチコアＣＰＵの各ＭＰＵコアに割当てる処理手順を示すフローチャート。

【図１２】監視制御プログラムモジュールが負荷分散制御指示行う際の動作を示すフローチャート。

【図１３】マルチコアＣＰＵの一部ＭＰＵコアに障害が発生した際の障害対応処理を表すフローチャート。

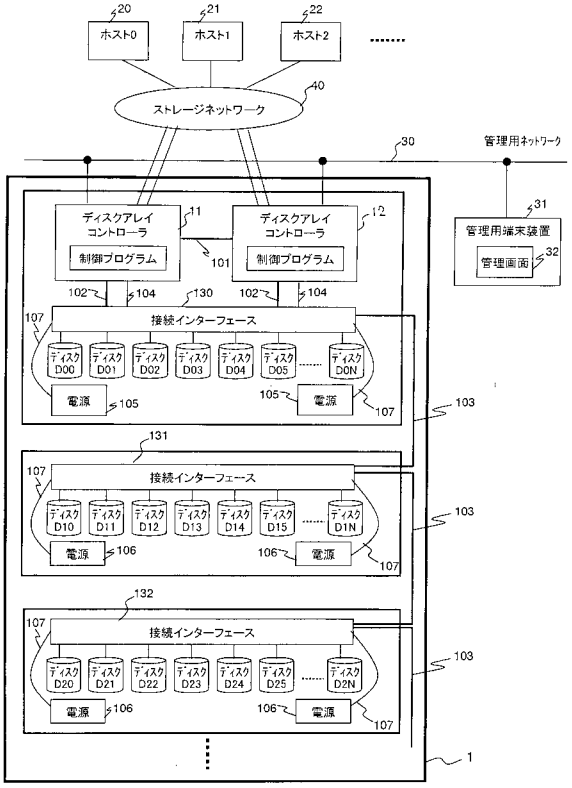
【符号の説明】

【０１０４】

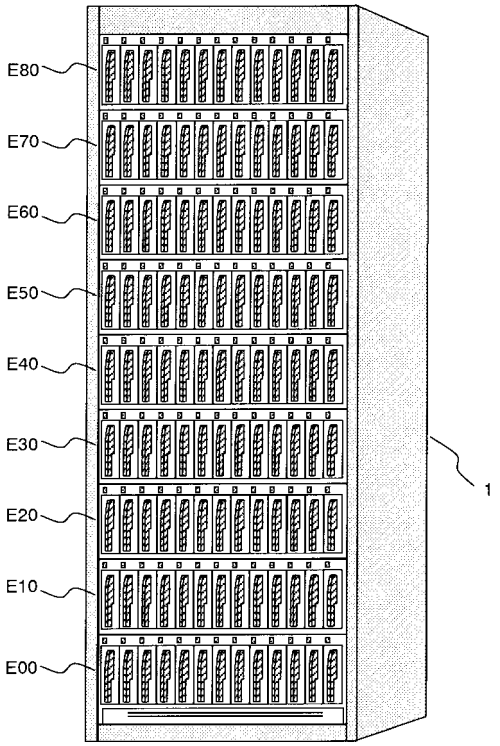
１…ディスクアレイ装置
１１、１２…ディスクアレイコントローラ
１０１、１０２、１０３…信号線
１０４、１０７…電力線
１０５、１０６…電源
１１０…複数ＭＰＵコア内蔵ＣＰＵ（マルチコアＣＰＵ）
１１１…不揮発性メモリ
１１２…揮発性メモリ
１１３…ホスト側接続ポート
１１４…ドライブ側接続ポート
１１５…管理用ネットワーク用物理ポート
１１６…ブートＲＯＭ
１１７…制御プログラム配置領域
１１８…データバッファ領域

１１９…制御プログラム
１１１０～１１１Ｎ…ＭＰＵコア
１１２０…ＣＰＵ内部スイッチ
１１３０…不揮発性メモリコントローラ
１１４０…ディスクアレイコントローラ間転送コントローラ
１１５０…パリティ演算器
１１６０…ＣＰＵ内部キャッシュ
１１７０…揮発性メモリコントローラ
１２０…信号線
１２２…接続コネクタ
１２３…ホスト側接続口
１２４…管理用ネットワーク接続口
１２５…エラーＬＥＤ
１３０、１３１、１３２…接続インターフェース
Ｄ００～Ｄ２Ｎ…ディスク装置
Ｅ００～Ｅ８０…ディスク筐体
２０、２１、２２…ホスト
３０…管理用ネットワーク
３１…管理用端末装置
３２…管理画面
４０…ストレージネットワーク
Ｐｒ１…監視制御プログラムモジュール
Ｐｒ２…運用管理プログラムモジュール
Ｐｒ３…ＲＡＩＤ制御プログラムモジュール
Ｐｒ４…ＮＡＳ制御プログラムモジュール
Ｐｒ５…ホストＩ／Ｏ制御プログラムモジュール
Ｐｒ６…ドライブＩ／Ｏ制御プログラムモジュール
Ｔｂ１…初期ＭＰＵコア割当管理テーブル
Ｔｂ２…初期閾値管理テーブル
Ｔｂ３…ＲＡＩＤ設定管理テーブル
Ｔｂ４…ＬＵ設定管理テーブル
Ｔｂ５…ＮＡＳボリューム管理テーブル
Ｔｂ６…ＭＰＵコア管理テーブル

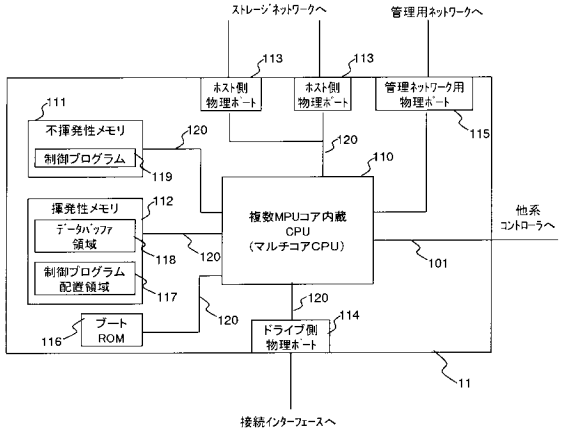
【図１】



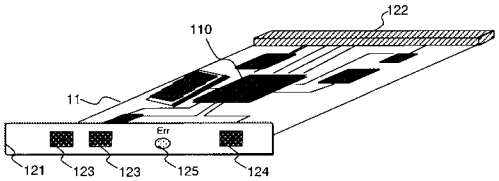
【図２】



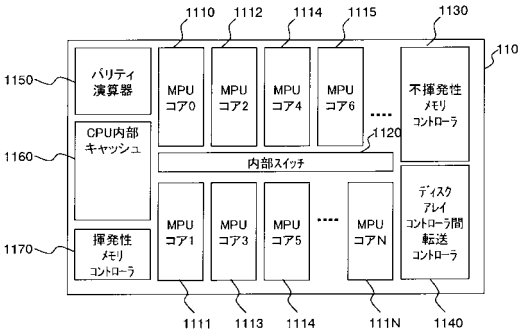
【図３】



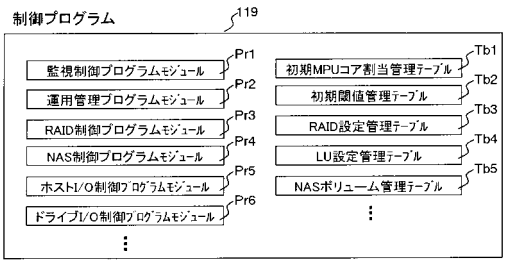
【図４】



【図５】



【図６】



【図７】

Tb1

MPU コアNo.	使用ﾓｼﾞｭｰﾙ	割当形式	特権ﾚﾍﾞﾙ	ｸﾞﾙｰﾌﾟNo
0	監視制御	固定	0	001
1	運用管理	固定	1	002
2	RAID制御	半固定	2	003
3	NAS制御	半固定	3	003
4	ﾎｽﾄI/O制御	半固定	4	004
5	ﾄﾞﾗｲﾌﾞI/O制御	半固定	4	004
6	(未割当)	動的	255	003
7	(未割当)	動的	255	003
8	(未割当)	動的	255	004
9	(未割当)	動的	255	-1

【図８】

Tb2

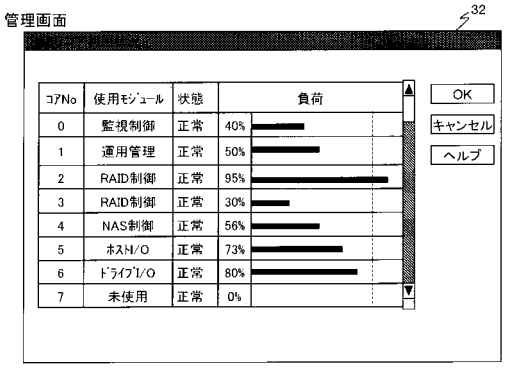
使用ﾓｼﾞｭｰﾙ	負荷閾値(%)	閾値オーバ規定回数
監視制御	100	1
運用管理	100	1
RAID制御	90	5
NAS制御	90	5
ﾎｽﾄI/O制御	90	5
ﾄﾞﾗｲﾌﾞI/O制御	90	5

【図９】

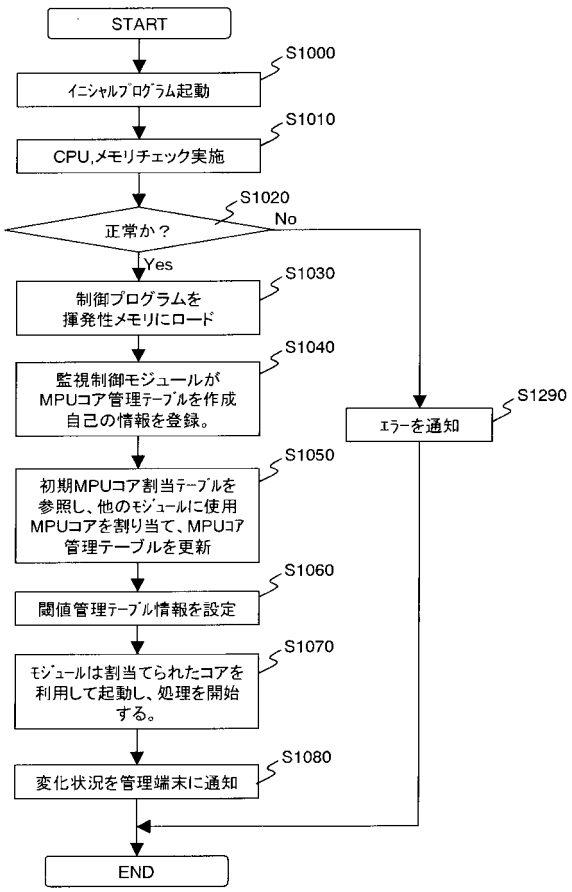
Tb6

MPU コアNo.	負荷(%)	ｸﾞﾙｰﾌﾟNo.	状態	使用ﾓｼﾞｭｰﾙ	割当形式	特権	負荷閾値ｵｰﾊﾞｰ回数
0	50%	001	正常	監視制御	固定	0	0
1	60%	002	正常	運用管理	固定	1	0
2	70%	003	正常	RAID制御	半固定	2	0
3	95%	003	正常	NAS制御	半固定	3	0
4	70%	003	正常	ﾎｽﾄI/O制御	半固定	4	0
5	40%	004	正常	ﾄﾞﾗｲﾌﾞI/O制御	半固定	4	0
6	0%	004	正常	(未割当)	動的	255	0
7	0%	004	正常	(未割当)	動的	255	0
8	0%	000	正常	(未割当)	動的	255	0

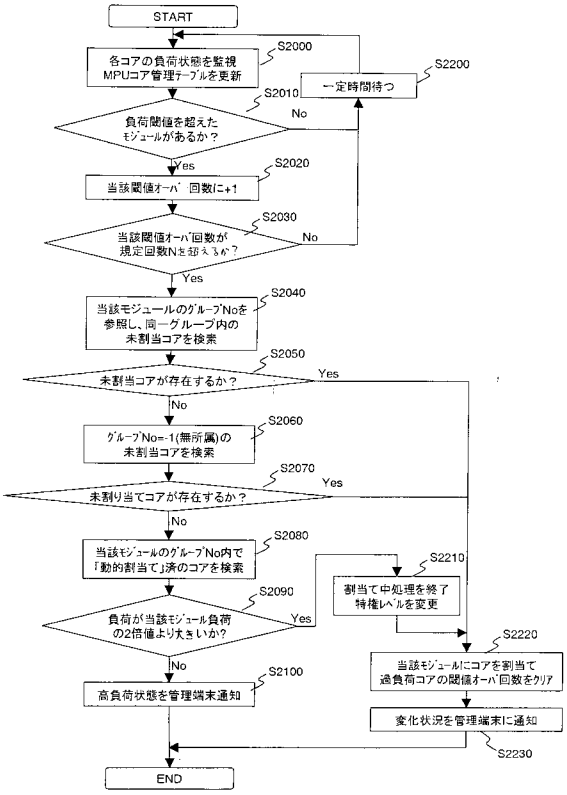
【図１０】



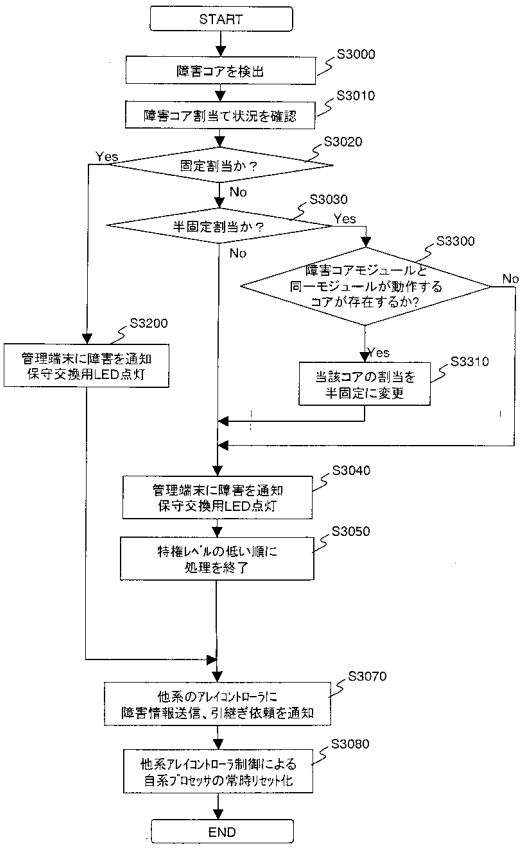
【図１１】



【図１２】


START
各コアの負荷状態を監視
MPUコア管理ﾃｰﾌﾞﾙを更新
S2000
一定時間待つ
S2200
負荷閾値を超えた
ﾓｼﾞｭｰﾙがあるか？
S2010
No
Yes
当該閾値ｵｰﾊﾞ 回数に+1
S2020
当該閾値ｵｰﾊﾞ回数が
規定回数Nを超えるか？
S2030
No
Yes
当該モジュールのｸﾞﾙｰﾌﾟNoを
参照し、同一グループ内の
未割当コアを検索
S2040
未割当コアが存在するか？
S2050
Yes
No
ｸﾞﾙｰﾌﾟNo=-1(無所属)の
未割当コアを検索
S2060
未割り当てコアが存在するか？
S2070
Yes
No
当該ﾓｼﾞｭｰﾙのｸﾞﾙｰﾌﾟNo内で
「動的割当て」済のコアを検索
S2080
負荷が当該ﾓｼﾞｭｰﾙ負荷
の2倍値より大きいか？
S2090
Yes
No
高負荷状態を管理端末通知
S2100
割当て中処理を終了
特権ﾚﾍﾞﾙを変更
S2210
当該ﾓｼﾞｭｰﾙにコアを割当て
過負荷コアの閾値ｵｰﾊﾞ回数をｸﾘｱ
S2220
変化状況を管理端末に通知
S2230
END

【図１３】


START
障害コアを検出
S3000
障害コア割当て状況を確認
S3010
固定割当か？
S3020
Yes
No
半固定割当か？
S3030
Yes
No
障害コアモジュールと
同一モジュールが動作する
コアが存在するか?
S3300
No
Yes
当該コアの割当を
半固定に変更
S3310
管理端末に障害を通知
保守交換用LED点灯
S3200
管理端末に障害を通知
保守交換用LED点灯
S3040
特権ﾚﾍﾞﾙの低い順に
処理を終了
S3050
他系のｱﾚｲｺﾝﾄﾛｰﾗに
障害情報送信、引継ぎ依頼を通知
S3070
他系ｱﾚｲｺﾝﾄﾛｰﾗ制御による
自系ﾌﾟﾛｾｯｻの常時ﾘｾｯﾄ化
S3080
END

フロントページの続き

(56)参考文献 特開平０２－２３８５５６（ＪＰ，Ａ）
特開２００１－１６７０４０（ＪＰ，Ａ）
特開平０４－２３５６６２（ＪＰ，Ａ）
特開平０９－２６５４３５（ＪＰ，Ａ）
特開２０００－１４８７０９（ＪＰ，Ａ）
特開２００２－３５３９６０（ＪＰ，Ａ）
特開２００５－０９９９８４（ＪＰ，Ａ）
特開平０９－２７４６０８（ＪＰ，Ａ）
特開平１１－０５３３２７（ＪＰ，Ａ）
特開２００３－２０８３６２（ＪＰ，Ａ）
特開平１１－３１６７２６（ＪＰ，Ａ）
特開２００５－２６６８４１（ＪＰ，Ａ）
枝廣　正人，マルチコア向けソフトウエア・プラットフォームを開発し携帯電話機に適用，日経エレクトロニクス，日本，日経ＢＰ社，２００５年　3月２８日，第896号，125～136頁
進藤　智則，シングルコアよりマルチコア　第１部＜発想の転換＞　迫るマイクロプロセサ危機　性能はコアの数で稼ぐ，日経エレクトロニクス，日本，日経ＢＰ社 Nikkei Business Publications,Inc.，２００４年　8月３０日，第881号，98～105頁
進藤　智則，シングルコアよりマルチコア　第３部＜ハードウエア＞　オンチップの恩恵生かす　高速バスや記憶階層がカギ，日経エレクトロニクス ，日本，日経ＢＰ社 Nikkei Business Publications,Inc.，２００４年　8月３０日，第881号，116～121頁
高務　祐哲　他，エネルギ最小周波数を利用したタスク再配置によるマルチプロセッサ向け消費エネルギ削減手法，電子情報通信学会技術研究報告　ＩＣＤ２００４－２３３～２４６　集積回路，日本，社団法人電子情報通信学会，２００５年　3月　4日，第104巻，第711号，37～42頁

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　　３／０６