

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号
特許第6074955号
(P6074955)

(45) 発行日 平成29年2月8日 (2017.2.8)

(24) 登録日 平成29年1月20日 (2017.1.20)

(51) Int. Cl.

F I

G O 6 F 11/20 (2006.01)

G O 6 F 11/20 6 3 3

G O 6 F 9/46 (2006.01)

G O 6 F 9/46 4 1 0

請求項の数 5 (全 25 頁)

(21) 出願番号	特願2012-191491 (P2012-191491)	(73) 特許権者	000005223
(22) 出願日	平成24年8月31日 (2012.8.31)		富士通株式会社
(65) 公開番号	特開2014-48907 (P2014-48907A)		神奈川県川崎市中原区上小田中4丁目1番1号
(43) 公開日	平成26年3月17日 (2014.3.17)	(74) 代理人	100092152
審査請求日	平成27年5月12日 (2015.5.12)		弁理士 服部 毅巖
		(72) 発明者	近藤 浩
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		(72) 発明者	安樂 隆史
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		審査官	井上 宏一
			最終頁に続く

(54) 【発明の名称】 情報処理装置および制御方法

(57) 【特許請求の範囲】

【請求項 1】

それぞれ処理を並列に実行する複数の個別回路を有する演算回路を、複数備えた情報処理装置において、

前記演算回路を単位として設定された使用権の数を記憶する記憶部と、

複数の前記演算回路のうち、前記使用権の数の前記演算回路の動作を許可するとともに、動作中の前記演算回路に搭載された前記複数の個別回路のうち第1の個別回路の異常を検知すると、動作していない前記演算回路が存在する場合には、前記第1の個別回路が搭載された前記演算回路における他の個別回路の動作を停止させ、動作していないいずれかの前記演算回路の動作を開始させ、動作していない前記演算回路が存在しない場合には、動作中の前記演算回路の中に使用されていない第2の個別回路があるかを判定し、前記第2の個別回路がある場合には、前記第1の個別回路を用いて実行されていた処理を、前記第2の個別回路を用いて実行させる制御部と、

を有することを特徴とする情報処理装置。

【請求項 2】

それぞれ処理を並列に実行する複数の個別回路を有する演算回路を、複数備えた情報処理装置において、

前記演算回路を単位として設定された使用権の数と、動作中の前記演算回路に搭載された前記複数の個別回路のうち、1つまたは複数のオペレーティングシステムのいずれかの処理に割り当てられた第1の個別回路の情報を保持する管理情報とを記憶する記憶部と、

複数の前記演算回路のうち、前記使用権の数の前記演算回路の動作を許可するとともに、前記第 1 の個別回路の異常を検知すると、動作していない前記演算回路が存在する場合には、前記第 1 の個別回路が搭載された前記演算回路における他の個別回路の動作を停止させ、動作していないいずれかの前記演算回路の動作を開始させ、動作していない前記演算回路が存在しない場合には、前記管理情報に基づき、動作中の前記演算回路に搭載された前記複数の個別回路の中にいずれのオペレーティングシステムの処理にも割り当てられていない第 2 の個別回路があるかを判定し、前記第 2 の個別回路がある場合には、前記第 1 の個別回路が前記第 2 の個別回路に置き換えられるように前記管理情報を更新する制御部と、

を有することを特徴とする情報処理装置。

10

【請求項 3】

前記情報処理装置は、複数のメモリ装置を有し、

前記記憶部は、前記メモリ装置を単位として設定されたメモリ使用権の数を記憶し、

前記制御部は、複数の前記メモリ装置のうち、前記メモリ使用権の数の前記メモリ装置の使用を許可するとともに、使用中の前記メモリ装置の記憶領域を分割した分割領域のいずれかから異常を検知すると、異常が検知された前記分割領域を含む前記メモリ装置における他の前記分割領域の使用を停止させ、使用されていないいずれかの前記メモリ装置の使用を開始する、

ことを特徴とする請求項 1 または 2 記載の情報処理装置。

【請求項 4】

20

それぞれ処理を並列に実行する複数の個別回路を有する演算回路を、複数備えた情報処理装置における制御方法であって、

前記演算回路を単位として設定された使用権の数を記憶する記憶部を参照して、複数の前記演算回路のうち、前記使用権の数の前記演算回路の動作を許可し、

動作中の前記演算回路に搭載された前記複数の個別回路のうち第 1 の個別回路の異常を検知すると、動作していない前記演算回路が存在する場合には、前記第 1 の個別回路が搭載された前記演算回路における他の個別回路の動作を停止させ、動作していないいずれかの前記演算回路の動作を開始させ、動作していない前記演算回路が存在しない場合には、動作中の前記演算回路の中に使用されていない第 2 の個別回路があるかを判定し、前記第 2 の個別回路がある場合には、前記第 1 の個別回路を用いて実行されていた処理を、前記

30

ことを特徴とする制御方法。

【請求項 5】

それぞれ処理を並列に実行する複数の個別回路を有する演算回路を、複数備えた情報処理装置における制御方法であって、

前記演算回路を単位として設定された使用権の数と、動作中の前記演算回路に搭載された前記複数の個別回路のうち、1 つまたは複数のオペレーティングシステムのいずれかの処理に割り当てられた第 1 の個別回路の情報を保持する管理情報とを記憶する記憶部を参照して、複数の前記演算回路のうち、前記使用権の数の前記演算回路の動作を許可し、

前記第 1 の個別回路の異常を検知すると、動作していない前記演算回路が存在する場合には、前記第 1 の個別回路が搭載された前記演算回路における他の個別回路の動作を停止させ、動作していないいずれかの前記演算回路の動作を開始させ、動作していない前記演算回路が存在しない場合には、前記管理情報に基づき、動作中の前記演算回路に搭載された前記複数の個別回路の中にいずれのオペレーティングシステムの処理にも割り当てられていない第 2 の個別回路があるかを判定し、前記第 2 の個別回路がある場合には、前記第 1 の個別回路が前記第 2 の個別回路に置き換えられるように前記管理情報を更新する、

40

ことを特徴とする制御方法。

【発明の詳細な説明】

【技術分野】

50

【 0 0 0 1 】

本発明は、情報処理装置および制御方法に関する。

【 背景技術 】

【 0 0 0 2 】

C o D (Capacity on Demand) は、サーバコンピュータなどの情報処理装置において、C P U (Central Processing Unit) などのハードウェア資源をユーザが必要な分だけ使用できるようにするものである。

【 0 0 0 3 】

例えば、C o D により C P U の使用数を制限する方法としては、ユーザは、複数の C P U が搭載された情報処理装置を購入する際、使用する C P U の数に応じた料金を支払う。これにより情報処理装置は、料金に応じた数の C P U が動作する一方、その数を越えた C P U の動作が禁止された状態で、ユーザに提供される。その後、情報処理装置の処理能力の増強が必要になった場合には、ユーザは、C P U の使用権を新たに購入することで、それまで使用していなかった C P U を直ちに使用可能になる。

10

【 0 0 0 4 】

C o D に関する技術の一例として、使用権が設定されたプロセッサが故障した場合に、使用権が設定されていない置き換えプロセッサを始動させて、故障プロセッサを停止させるようにしたものがある。

【 0 0 0 5 】

また、近年のマルチコアプロセッサの中には、プロセッサ内の各コアの内部に複数のハードウェアスレッドを備えるものがある。ハードウェアスレッドは「ストランド (Strand)」と呼ばれることもある。コア内のハードウェアスレッドは、それぞれ個別の処理を並列に実行可能である。また、ハードウェアスレッドごとにレジスタなどのハードウェア資源が個別に設けられることもある。

20

【 先行技術文献 】

【 特許文献 】

【 0 0 0 6 】

【 特許文献 1 】 特開 2 0 0 4 - 2 8 8 1 8 3 号 公 報

【 特許文献 2 】 特開 2 0 0 4 - 3 1 8 8 8 5 号 公 報

【 特許文献 3 】 国際公開第 2 0 0 7 / 0 9 9 6 0 6 号

30

【 発明の概要 】

【 発明が解決しようとする課題 】

【 0 0 0 7 】

ところで、上記のように複数のハードウェアスレッドを備えたコアを複数含むマルチコアプロセッサが搭載された情報処理装置において、C o D によりコアごとに使用権を設定した場合には、次のような問題がある。ハードウェアスレッドごとに個別に設けられたハードウェアのいずれかに異常が発生したときに、異常が発生したハードウェアスレッドを別のハードウェアスレッドに置き換えると、その置き換えの仕方によってはライセンス超過が発生する場合がある。

40

【 0 0 0 8 】

1 つの側面では、本発明は、異常が発生したハードウェアの置き換え時におけるライセンス超過の発生を防止した情報処理装置および制御方法を提供することを目的とする。

【 課題を解決するための手段 】

【 0 0 0 9 】

1 つの案では、複数の処理を並列に実行可能であって、並列に実行する処理ごとに個別に設けられた個別回路を有する演算回路を、複数備えた情報処理装置が提供される。この情報処理装置は、次のような記憶部および制御部を有する。記憶部は、演算回路を単位として設定された使用権の数を記憶する。制御部は、複数の演算回路のうち、使用権の数の演算回路の動作を許可する。また、制御部は、動作中の演算回路に搭載されたいずれかの個別回路の異常を検知すると、異常が検知された個別回路が搭載された演算回路における

50

他の個別回路の動作を停止させ、動作していないいずれかの演算回路の動作を開始させる。

【 0 0 1 0 】

また、1つの案では、上記の情報処理装置と同様の処理を実行する制御方法が提供される。

【発明の効果】

【 0 0 1 1 】

1態様によれば、情報処理装置および制御方法において、異常が発生したハードウェアの置き換え時におけるライセンス超過の発生を防止できる。

【図面の簡単な説明】

【 0 0 1 2 】

【図1】第1の実施の形態に係る情報処理装置の構成例およびその動作例を示す図である。

【図2】第2の実施の形態に係るサーバ装置の全体構成例を示す図である。

【図3】CPUチップの内部構成例を示す図である。

【図4】スレッドの故障が発生した場合の処理の参考例を示す図である。

【図5】第2の実施の形態におけるスレッド故障時の処理例を示す図である。

【図6】情報処理装置が備える機能の構成例を示すブロック図である。

【図7】コア管理テーブルに登録される情報の例を示す図である。

【図8】ドメイン管理テーブルに登録される情報の例を示す図である。

【図9】スレッドの置き換え処理の例を示すフローチャートである。

【図10】スレッドの置き換え処理の例を示すフローチャートである。

【図11】スレッド切り離し処理の例を示すフローチャートである。

【図12】スレッド組み込み処理の例を示すフローチャートである。

【図13】コア管理テーブルの遷移の例を示す図である。

【図14】ドメイン管理テーブルの遷移の例を示す図である。

【図15】第3の実施の形態に係る情報処理装置が備える機能の構成例を示すブロック図である。

【図16】メモリ装置に対するライセンス設定の第1の例を示す図である。

【図17】メモリ装置に対するライセンス設定の第2の例を示す図である。

【発明を実施するための形態】

【 0 0 1 3 】

以下、本発明の実施の形態について図面を参照して説明する。

〔第1の実施の形態〕

図1は、第1の実施の形態に係る情報処理装置の構成例およびその動作例を示す図である。図1に示す情報処理装置1は、記憶部2、制御部3および複数の演算回路を有している。図1の例では、情報処理装置1は2つの演算回路11, 12を備えているが、演算回路の搭載数は2つ以上の任意の数とすることができる。

【 0 0 1 4 】

演算回路11, 12は、例えば、プロセッサの内部に搭載されるプロセッサコア、またはプロセッサとして実現される。制御部3も、例えば、プロセッサコアまたはプロセッサとして実現される。制御部3は、例えば、複数の演算回路のうちの1つとして実現されてもよい。記憶部2は、HDD (Hard Disk Drive)、SSD (Solid State Drive) などの不揮発性記憶装置として実現される。

【 0 0 1 5 】

演算回路11, 12のそれぞれは、複数の処理を並列に実行可能である。また、演算回路11, 12のそれぞれは、並列に実行する処理ごとに個別に設けられた個別回路を有する。演算回路11, 12がプロセッサコアまたはプロセッサである場合、個別回路は、例えば、レジスタセット、命令バッファ、演算器、またはこれらの任意の組み合わせとして実現される。

10

20

30

40

50

【 0 0 1 6 】

本実施の形態では例として、図 1 の右側に示すように、演算回路 1 1 は個別回路 2 1 , 2 2 を有し、演算回路 1 2 は個別回路 2 3 , 2 4 を有するものとする。すなわち、演算回路 1 1 は、個別回路 2 1 , 2 2 を用いて 2 系統の処理を並列に実行でき、演算回路 1 2 は、個別回路 2 3 , 2 4 を用いて 2 系統の処理を並列に実行できる。なお、「複数の処理を並列に実行する」とは、各処理を同時に実行することの他、各処理を時分割して交互に実行することや、各処理をパイプライン方式で実行することを含んでもよい。

【 0 0 1 7 】

情報処理装置 1 には、演算回路を単位として使用権が設定される。使用権は、例えば、情報処理装置 1 のユーザが所定の料金を支払うことで設定される。記憶部 2 は、設定された使用権の数を記憶する。

10

【 0 0 1 8 】

制御部 3 は、記憶部 2 に記憶された使用権の数の演算回路の動作を許可する。例えば、使用権の数が「1」である場合、制御部 3 は、演算回路 1 1 , 1 2 の一方の動作を許可し、他方の動作を禁止する。これにより、ユーザは、情報処理装置 1 に演算回路が 2 つ以上搭載されている場合であっても、1 つの演算回路だけしか使用できない。

【 0 0 1 9 】

また、制御部 3 は、動作中の演算回路に搭載されたいずれかの個別回路の異常を検知したとき、異常が検知された個別回路が搭載された演算回路に含まれる他の個別回路の動作を停止させ、動作していないいずれかの演算回路の動作を開始させる。ここで、制御部 3 は、個別回路の異常を検知したとき、異常が検知された個別回路が搭載された演算回路における他の少なくとも 1 つの個別回路から異常が検知されていなくても、その演算回路内の他のすべての動作を停止する。このような処理により、動作中の演算回路の数が使用権の数より多くなるというライセンス超過の発生が防止される。

20

【 0 0 2 0 】

例えば、図 1 に「状態 1」として示すように、演算回路 1 1 が動作中であり、演算回路 1 2 が動作していないものとする。また、設定された使用権の数が「1」であるものとする。なお、図 1 の右側では、動作中の回路を実線で示し、動作していない回路を点線で示している。

【 0 0 2 1 】

ここで、制御部 3 が個別回路 2 1 での異常発生を検知したものとする。このとき制御部 3 は、個別回路 2 2 の異常を検知していなくても、個別回路 2 2 の動作を停止させる。これにより、演算回路 1 1 自体の動作が停止する。そして、制御部 3 は、動作していなかった演算回路 1 2 の動作を開始させる。このように動作する演算回路が置き換えられた状態を、図 1 に「状態 2」として示す。この状態において制御部 3 は、例えば、個別回路 2 1 を用いて実行されていた処理を、個別回路 2 3 を用いて引き続き実行させ、個別回路 2 2 を用いて実行されていた処理を、個別回路 2 4 を用いて引き続き実行させる。

30

【 0 0 2 2 】

仮に、個別回路 2 1 の異常を検知したとき、個別回路 2 1 の動作が停止されて、演算回路 1 2 内の個別回路 2 3 , 2 4 のいずれかの動作が開始された場合には、演算回路 1 2 もユーザに使用されることになる。この場合、使用される演算回路の数が「2」になって、ライセンス超過が発生してしまう。

40

【 0 0 2 3 】

これに対して、本実施の形態の制御部 3 の処理によれば、個別回路の異常検知時に必ず演算回路を単位としてハードウェアの置き換えが行われることから、ライセンス超過の発生が防止される。図 1 の右側の動作例では、ハードウェアの置き換え前を示す状態 1 と、置き換え後を示す状態 2 では、いずれも動作中の演算回路の数は「1」であり、設定された使用権の数の範囲に収まっている。

【 0 0 2 4 】

〔 第 2 の実施の形態 〕

50

次に、情報処理装置の例として、マルチコア・マルチスレッド方式のプロセッサが搭載されたサーバ装置を挙げて説明する。

【 0 0 2 5 】

図 2 は、第 2 の実施の形態に係るサーバ装置の全体構成例を示す図である。図 2 に示すサーバ装置 1 0 0 は、ビルディングブロック 1 1 0 a ~ 1 1 0 d およびサービスプロセッサ 1 2 0 を備える。ビルディングブロック 1 1 0 a ~ 1 1 0 d およびサービスプロセッサ 1 2 0 は、例えば、ラック内のスロットにそれぞれ挿入され、ラックのバックボードに設けられたバス 1 0 1 を通じて互いに接続されて、データを送受信できるようになっている。

【 0 0 2 6 】

ビルディングブロック 1 1 0 a は、CPU チップ 1 1 1 a , 1 1 1 b、メモリ 1 1 2 a , 1 1 2 b、ディスクユニット 1 1 3 および通信インタフェース 1 1 4 を備える。ビルディングブロック 1 1 0 a 内のこれらのハードウェア要素は、例えば、図示しないクロスバーを介して互いに接続される。また、ディスクユニット 1 1 3 には、HDD 1 1 5 a , 1 1 5 b が搭載されている。

【 0 0 2 7 】

CPU チップ 1 1 1 a , 1 1 1 b はそれぞれ、後述するように複数の CPU コアを備えた、マルチコア方式のプロセッサである。CPU チップ 1 1 1 a は、ディスクユニット 1 1 3 内の HDD 1 1 5 a , 1 1 5 b などの不揮発性記憶装置からプログラムを読み出してメモリ 1 1 2 a , 1 1 2 b の少なくとも一方にロードし、ロードしたプログラムを実行する。CPU チップ 1 1 1 b も同様に、ディスクユニット 1 1 3 内の HDD 1 1 5 a , 1 1 5 b などの不揮発性記憶装置からプログラムを読み出してメモリ 1 1 2 a , 1 1 2 b の少なくとも一方にロードし、ロードしたプログラムを実行する。

【 0 0 2 8 】

メモリ 1 1 2 a , 1 1 2 b は、ともに揮発性の記憶装置である。メモリ 1 1 2 a には、CPU チップ 1 1 1 a , 1 1 1 b の少なくとも一方に実行させるプログラムの少なくとも一部や、プログラムの実行に必要な各種データが一時的に格納される。メモリ 1 1 2 b にも同様に、CPU チップ 1 1 1 a , 1 1 1 b の少なくとも一方に実行させるプログラムの少なくとも一部や、プログラムの実行に必要な各種データが一時的に格納される。

【 0 0 2 9 】

ディスクユニット 1 1 3 内の HDD 1 1 5 a , 1 1 5 b には、プログラムや各種データが格納される。なお、ディスクユニット 1 1 3 には、SSD などの他の種類の不揮発性記憶装置が搭載されてもよい。

【 0 0 3 0 】

通信インタフェース 1 1 4 は、バス 1 0 1 に接続し、バス 1 0 1 と CPU チップ 1 1 1 a , 1 1 1 b との間でデータを送受信する。

なお、ビルディングブロック 1 1 0 a に搭載される CPU チップの数や RAM の数、およびディスクユニット 1 1 3 に搭載される不揮発性記憶装置の数は、それぞれ任意の数とすることができる。

【 0 0 3 1 】

ビルディングブロック 1 1 0 b ~ 1 1 0 d のそれぞれは、ビルディングブロック 1 1 0 a と同様のハードウェア構成を有しており、ここでは説明を省略する。

以上のように、サーバ装置 1 0 0 は、それぞれに 1 つ以上のプロセッサ（ここでは CPU チップを指す）が搭載されたビルディングブロックを複数備える。このような構成により、サーバ装置 1 0 0 全体としてマルチプロセッサシステムが実現される。

【 0 0 3 2 】

サービスプロセッサ 1 2 0 は、ビルディングブロック 1 1 0 a ~ 1 1 0 d の動作状態を監視する。サービスプロセッサ 1 2 0 は、CPU 1 2 1、メモリ 1 2 2、HDD 1 2 3 および通信インタフェース 1 2 4 を備える。

【 0 0 3 3 】

10

20

30

40

50

CPU 121は、HDD 123に格納されたファームウェアプログラムをメモリ 122にロードし、ロードしたファームウェアプログラムを実行する。これによりCPU 121は、ビルディングブロック 110a ~ 110dの監視を含む所定の処理を実行する。

【0034】

メモリ 122は、ファームウェアプログラムの少なくとも一部や各種データを一時的に記憶する揮発性記憶装置である。HDD 123は、不揮発性記憶装置の一例であり、例えば、SSDなどの他の種類と置換されてもよい。通信インタフェース 124は、バス 101に接続し、バス 101とCPU 121との間でデータを送受信する。

【0035】

なお、図2のサーバ装置 100には例として4台のビルディングブロックが搭載されているが、ビルディングブロックの搭載数は任意の数とすることができる。また、図2のサーバ装置 100には例として1台のサービスプロセッサが搭載されているが、複数台のサービスプロセッサが搭載されていてもよい。例えば、動作の信頼性向上のためにサービスプロセッサが冗長化されていてもよい。

【0036】

図3は、CPUチップの内部構成例を示す図である。この図3に示すCPUチップ 111は、ビルディングブロック 110a ~ 110dのそれぞれに搭載されたCPUチップ（例えば、図2のCPUチップ 111aまたはCPUチップ 111b）に対応する。すなわち、ビルディングブロック 110a ~ 110dはそれぞれ、図3と同様の構成のCPUチップを備えている。

【0037】

CPUチップ 111は、コア 201a ~ 201dおよび二次キャッシュ 202を備えている。コア 201a ~ 201dは、それぞれ第1の実施の形態における演算回路に対応する。コア 201a ~ 201dは、それぞれ個別の演算器を少なくとも備えており、それらの演算器によってそれぞれ個別の処理を並列に実行可能である。なお、CPUチップ 111に搭載されるコアの数は、任意の数とすることができる。二次キャッシュ 202は、コア 201a ~ 201dに共用されるキャッシュメモリである。

【0038】

コア 201aは、スレッド 210a, 210bおよび一次キャッシュ 220を備えている。なお、コア 201a ~ 201dはそれぞれ同様の構成であるので、ここでは代表してコア 201aについてのみ説明する。

【0039】

一次キャッシュ 220は、スレッド 210a, 210bに共用されるキャッシュメモリである。一次キャッシュ 220は、命令をキャッシュする命令キャッシュ 221と、データをキャッシュするデータキャッシュ 222とを備える。

【0040】

スレッド 210a, 210bは、それぞれ「ストランド」とも呼ばれるハードウェアスレッドである。スレッド 210a, 210bは、それぞれ個別の処理を並列に実行可能である。ここで言う個別の処理とは、例えば、OS (Operating System) 上でプログラムが実行される際に、そのOS自身が認識しているCPUごとに実行される単位処理（例えば、ソフトウェアの意味でのスレッド）である。

【0041】

スレッド 210a, 210bに対応するハードウェアの少なくとも一部は、スレッド 210aのためのものとスレッド 210bのためのものとで個別に搭載されている。そして、スレッド 210a, 210bは、それぞれ個別の処理を実行する際に、自分自身に対応する個別のハードウェアを利用する。

【0042】

図3の例では、スレッド 210aは、自分専用のハードウェアとして命令バッファ 212aおよびレジスタ部 214aを備える。スレッド 210bも同様に、自分専用のハードウェアとして命令バッファ 212bおよびレジスタ部 214bを備える。一方、命令制御

10

20

30

40

50

部 2 1 1 および演算部 2 1 3 は、スレッド 2 1 0 a とスレッド 2 1 0 b とで共通化されている。

【 0 0 4 3 】

なお、命令バッファ 2 1 2 a およびレジスタ部 2 1 4 a は、第 1 の実施の形態における個別回路の 1 つに対応し、命令バッファ 2 1 2 b およびレジスタ部 2 1 4 b は、第 1 の実施の形態における個別回路の他の 1 つに対応する。

【 0 0 4 4 】

命令制御部 2 1 1 は、メモリから命令を読み込む。ここで言うメモリとは、命令キャッシュ 2 2 1、二次キャッシュ 2 0 2、CPU チップ 1 1 1 内の主記憶装置（図 2 のメモリ 1 1 2 a、1 1 2 b に対応）のいずれかである。また、命令制御部 2 1 1 は、メモリから読み込んだ命令を命令バッファ 2 1 2 a、2 1 2 b のいずれかに一時的に格納した後、格納した命令を読み込む場合もある。スレッド 2 1 0 a に対応する処理では命令バッファ 2 1 2 a が使用され、スレッド 2 1 0 b に対応する処理では命令バッファ 2 1 2 b が使用される。

10

【 0 0 4 5 】

命令制御部 2 1 1 は、読み込んだ命令をデコードし、デコード結果に応じた演算処理を演算部 2 1 3 に実行させる。演算部 2 1 3 が実行する演算処理としては、例えば、加算、減算、乗算、除算、アドレス計算などがある。演算部 2 1 3 は、演算に使用するデータの読み出しや、演算結果を書き込むために、レジスタ部 2 1 4 a、2 1 4 b のいずれかのレジスタを利用する。また、ロード命令やストア命令が発行された場合、メモリとレジスタ部 2 1 4 a またはレジスタ部 2 1 4 b との間でデータが転送される。ここで言うメモリとは、データキャッシュ 2 2 2、二次キャッシュ 2 0 2、CPU チップ 1 1 1 内の主記憶装置（図 2 のメモリ 1 1 2 a、1 1 2 b に対応）のいずれかである。このような演算部 2 1 3 による処理においては、スレッド 2 1 0 a に対応する処理ではレジスタ部 2 1 4 a が使用され、スレッド 2 1 0 b に対応する処理ではレジスタ部 2 1 4 b が使用される。

20

【 0 0 4 6 】

このような構成のコア 2 0 1 a によれば、例えば、一方のスレッドに対応する演算部 2 1 3 での演算処理と、他方のスレッドに対応する、メモリと命令バッファまたはレジスタ部とのデータ転送処理とを、並列に実行することができる。これにより、処理性能が向上する。

30

【 0 0 4 7 】

また、命令制御部および演算部も、スレッドごとに個別に設けられてもよい。この場合、コア 2 0 1 a は、各スレッドに対応する演算処理を同時に実行することができる。

ところで、本実施の形態のサーバ装置 1 0 0 は、C o D 技術に対応している。サーバ装置 1 0 0 のユーザは、コアを単位としてライセンス（使用权）を購入する。サーバ装置 1 0 0 は、ライセンスが購入された台数分のコアを動作許可状態にし、その台数に含まれない残りのコアを動作禁止状態にする。また、ユーザがライセンスを追加購入した場合には、動作禁止状態のコアのうち、追加購入された台数分のコアが動作許可状態に変更される。これにより、ユーザは、サーバ装置 1 0 0 を導入する際の初期投資を抑制し、さらにその後、必要に応じてサーバ装置 1 0 0 の能力を即座に増強することができる。

40

【 0 0 4 8 】

ここで、ライセンスの単位が、コア内のスレッドではなくコアとされているのは、次のような理由による。図 3 に示したように、コア内の各スレッドは、コア内のハードウェアの一部を共用する。例えば図 3 の場合、一次キャッシュ 2 2 0、命令制御部 2 1 1 および演算部 2 1 3 が、スレッド 2 1 0 a、2 1 0 b によって共用される。このため、コア内の一方のスレッドのみが動作した場合と、コア内の両方のスレッドが動作した場合とでは、後者の方がスレッド当たりの処理速度が低下する場合がある。

【 0 0 4 9 】

このことから、スレッドごとに一定のライセンス料金を設定してしまうと、スレッドの動作状況に応じて、ライセンス料金と処理速度とのバランスが変動してしまうことになり

50

、料金の設定の仕方が適切とは言えなくなる。これに対して、CPUチップ内の複数のコアに関しては、各コアの動作状況によるコア当たりの処理速度の変動は発生しにくい。このため、ライセンス料金と処理速度とのバランスをできるだけ保つためには、ライセンスの最小単位をスレッドではなくコアとする方が好ましい。

【0050】

次に、スレッド内のハードウェアで故障が発生した場合の処理について説明する。以下の説明では、スレッドごとに個別に設けられたハードウェアで故障が発生した場合を想定する。

【0051】

また、以下の説明において、「スレッドの故障」とは、スレッドのハードウェアのうち、スレッドごとに個別に設けられたハードウェアで故障が発生したことを指すものとする。例えば図3において、「スレッド210aの故障」とは、スレッド210a内の命令バッファ212aまたはレジスタ部214aの少なくとも一方で故障が発生したことを指す。スレッドの故障の具体例としては、いずれかのスレッドにおいてレジスタのパリティエラーが発生するケースなどがある。

【0052】

さらに、以下の説明では、識別番号「x」を「#x」と表記する。例えば、識別番号が「0」のスレッドを「スレッド#0」と表記する。

図4は、スレッドの故障が発生した場合の処理の参考例を示す図である。なお、この図4、および次の図5では、動作許可状態のハードウェアを実線で示し、動作禁止状態のハードウェアを点線で示す。

【0053】

図4では例として、コア3台分のライセンスが設定されているものとする。そして、このようなライセンスの設定状況に従い、図4の上段に示すように、CPUチップ111a内の4つのコアのうち、コア#0、#1、#2が動作中であり、コア#3が動作していないものとする。

【0054】

ここで例として、サーバ装置100は、コア#1のスレッド#2から故障を検知したものとする。この場合、サーバ装置100は、故障が検知されたスレッド#2を、動作していない他のスレッドに置き換えて、スレッド#2で実行されていた処理を置き換え後のスレッドに引き継がせる。

【0055】

スレッドの置き換え方法としては、故障が検知されたスレッド#2を、他のコア内の動作していないスレッドに置き換える方法が考えられる。図4の下段は、故障が検知されたスレッド#2を、コア#3内の動作していないスレッド#6に置き換えた例を示している。しかしながら、この方法では、スレッド#3を用いた処理はそのまま続行されるので、コア#1が使用された状態のままコア#3が新たに使用されることになる。これにより、使用されるコアの数が「3」から「4」に増加してしまい、ライセンス超過の状態になってしまう。

【0056】

これに対して、本実施の形態では、次の図5に示すようなハードウェアの置き換え処理が行われることで、ライセンス超過の発生が防止される。図5は、第2の実施の形態におけるスレッド故障時の処理例を示す図である。

【0057】

図5の上段は、図4の上段と同様の状態を示しており、この状態からコア#1のスレッド#2から故障が検知されたものとする。このとき、サーバ装置100は、コア#1内の他のスレッド#3から故障が検知されていなくても、コア#1を動作禁止状態にする。これとともに、サーバ装置100は、動作していない他のいずれかのコアの動作を開始させる。図5の例では、動作していなかったコア#3の動作が開始され、図5の下段はこのときの状態を示す。この場合、サーバ装置100は、スレッド#2で実行されていた処理を

10

20

30

40

50

、例えばスレッド# 6 に引き継がせ、スレッド# 3 で実行されていた処理を、例えばスレッド# 7 に引き継がせる。

【 0 0 5 8 】

このような置き換え処理によれば、置き換え処理の前後で、動作中のコアの数は「 3 」のまま変化せず、ライセンス超過は発生しない。

次に、サーバ装置 1 0 0 の処理の詳細について説明する。

【 0 0 5 9 】

まず、図 6 は、情報処理装置が備える機能の構成例を示すブロック図である。サーバ装置 1 0 0 は、ハイパーバイザ 1 3 0 およびドメイン管理部 1 4 0 を備える。また、サーバ装置 1 0 0 の記憶装置には、コア管理テーブル 1 5 1 およびドメイン管理テーブル 1 5 2 が記憶される。

10

【 0 0 6 0 】

ここで、ハイパーバイザ 1 3 0 の処理は、サーバ装置 1 0 0 が備えるプロセッサがハイパーバイザプログラムを実行することで実現される。また、ドメイン管理部 1 4 0 の処理は、サーバ装置 1 0 0 が備えるプロセッサがドメイン管理プログラムを実行することで実現される。ハイパーバイザプログラムおよびドメイン管理プログラムは、例えば、いずれかのビルディングブロックに搭載された少なくとも 1 つの CPU チップによって実行される。

【 0 0 6 1 】

サーバ装置 1 0 0 は、1 つ以上の仮想マシンを構築する。図 6 に示す論理ドメイン # 0 ~ # 3 は、それぞれが仮想マシンに対応し、論理ドメイン # 0 ~ # 3 のそれぞれには、仮想的なハードウェア資源として CPU、メモリ、ディスク（不揮発性記憶装置）などが割り当てられる。ここで、各論理ドメインに割り当てられる仮想 CPU は、コア内のスレッドに対応する。また、各論理ドメインには、複数の仮想 CPU が割り当てられてもよい。

20

【 0 0 6 2 】

そして、各論理ドメインにおいては、割り当てられた 1 つ以上の仮想 CPU、すなわち 1 つ以上のスレッドが、ゲスト OS プログラムを実行する。これにより、各論理ドメインにおいてそれぞれ個別のゲスト OS が動作する。図 6 の例では、論理ドメイン # 0 , # 1 , # 2 , # 3 において、それぞれゲスト OS # 0 , # 1 , # 2 , # 3 が動作する。

【 0 0 6 3 】

ハイパーバイザ 1 3 0 は、ゲスト OS # 0 ~ # 3 に対して、実ハードウェアへのインタフェースを提供する。これによりゲスト OS # 0 ~ # 3 は、ハイパーバイザ 1 3 0 上で動作する。

30

【 0 0 6 4 】

また、ハイパーバイザ 1 3 0 は、ハードウェアの異常を検知する異常検知部 1 3 1 を備える。異常検知部 1 3 1 は、少なくとも、コア内のスレッドごとに個別のハードウェア（本実施の形態では命令バッファおよびレジスタ部）の異常発生を検知することができる。

【 0 0 6 5 】

ドメイン管理部 1 4 0 は、コア管理テーブル 1 5 1 およびドメイン管理テーブル 1 5 2 を利用して、各論理ドメインに対する実ハードウェアの割り当てを管理する。コア管理テーブル 1 5 1 には、コアのライセンスに関する情報が登録される。また、ドメイン管理テーブル 1 5 2 には、論理ドメインごとのスレッドの割り当て状況に関する情報が登録される。

40

【 0 0 6 6 】

また、ドメイン管理部 1 4 0 は、ゲスト OS の動作を停止させずに、ゲスト OS に対応する論理ドメインからのハードウェア資源の切り離しや、論理ドメインへのハードウェア資源の組み込みを行うための「動的再構成 (Dynamic Reconfiguration)」の制御を行う。ドメイン管理部 1 4 0 は、スレッドの異常が検知された場合に、異常が検知されたスレッドが割り当てられているゲスト OS に対して、スレッドの切り離しおよび新たなスレッドの組み込みを指示する。

50

【 0 0 6 7 】

図 7 は、コア管理テーブルに登録される情報の例を示す図である。コア管理テーブル 1 5 1 には、ライセンスありのコア数、ライセンスなしのコア数、および異常コア数が登録される。

【 0 0 6 8 】

ライセンスありのコア数は、ユーザによって購入されたコアのライセンス数を示す。コア管理テーブル 1 5 1 において、ライセンスありのコア数の欄は、さらに、ドメイン割り当て済みコア数と、ドメイン未割り当てのコア数とに分けられている。ドメイン割り当て済みのコア数は、論理ドメインに割り当てられているコアの数を示し、ドメイン未割り当てのコア数は、論理ドメインに割り当てられていないコアの数を示す。ドメイン割り当て済みのコア数およびドメイン未割り当てのコア数の各欄に登録された数の合計値が、ユーザによって購入されたライセンスの総数を示す。

10

【 0 0 6 9 】

異常コア数は、サーバ装置 1 0 0 に搭載されたコアのうち、異常が検知されて他のコアに置き換えられたコアの数を示す。ライセンスなしのコア数は、サーバ装置 1 0 0 に搭載されたコアの総数から、購入されたライセンスの数と、異常コア数とを減算した数を示す。従って、コア管理テーブル 1 5 1 のすべての欄に登録されたコア数を合計した値は、サーバ装置 1 0 0 に搭載された、ライセンス設定対象のコアの総数を示す。

【 0 0 7 0 】

サーバ装置 1 0 0 が出荷される段階では、例えば、ドメイン未割り当てのコア数の欄に、ユーザによって購入されたライセンスの数が登録される。これとともに、ライセンスなしのコア数の欄に、ライセンス設定対象のコアの総数から、購入されたライセンス数を減算した値が登録される。このとき、ドメイン割り当て済みのコア数および異常コア数の各欄には、「 0 」が登録される。

20

【 0 0 7 1 】

ドメイン管理部 1 4 0 は、例えばサーバ装置 1 0 0 に接続された管理端末（図示せず）からの入力に応じて、論理ドメインの設定を行う。ドメイン管理部 1 4 0 は、論理ドメインにスレッドを割り当てする際、そのスレッドを含むコア内の他のいずれのスレッドも論理ドメインに割り当てられていない場合には、ドメイン割り当て済みのコア数に「 1 」を加算するとともに、ドメイン未割り当てのコア数から「 1 」を減算する。

30

【 0 0 7 2 】

また、ユーザによってライセンスが追加購入されたとき、ドメイン管理部 1 4 0 は、追加されたライセンス数の入力を受ける。ドメイン管理部 1 4 0 は、入力された追加ライセンス数を、ドメイン未割り当てのコア数に加算するとともに、同じ追加ライセンス数を、ライセンスなしのコア数から減算する。これにより、ユーザが使用できるコアの数が追加ライセンスの数だけ増加する。

【 0 0 7 3 】

さらに、ドメイン管理部 1 4 0 は、スレッドの異常を検知したとき、上記のように、異常が検知されたスレッドを含むコアを、他のコアに置き換える。異常が検知されたスレッドがいずれかの論理ドメインに割り当て済みであった場合、ドメイン管理部 1 4 0 は、ドメイン割り当て済みのコア数から「 1 」を減算するとともに、異常コア数に「 1 」を加算する。

40

【 0 0 7 4 】

図 8 は、ドメイン管理テーブルに登録される情報の例を示す図である。ドメイン管理テーブル 1 5 2 には、コア ID およびスレッド ID に対応付けて、ステータスおよびドメイン ID が登録される。

【 0 0 7 5 】

コア ID は、ライセンス設定対象のコアを識別するための情報である。スレッド ID は、対応するコアに含まれるスレッドを識別するための情報である。

ステータスは、対応するスレッドの状態を示す。ステータスの欄には、「 in use 」 , 「

50

free」,「fail」のいずれかが登録される。「in use」は、対応するスレッドに論理ドメインが割り当てられていることを示す。「free」は、対応するスレッドに論理ドメインが割り当てられていないことを示す。「fail」は、対応するスレッドから異常が検出され、そのスレッドに対する論理ドメインの割り当てが禁止されていることを示す。

【0076】

ドメイン管理部140は、2つのスレッドのうち少なくとも一方のステータスが「in use」であるコアを、コア管理テーブル151におけるドメイン割り当て済みのコアの1つとしてカウントする。また、2つのスレッドのステータスがいずれも「free」であるコアは、コア管理テーブル151におけるドメイン未割り当てのコアまたはライセンスなしのコアのいずれかとしてカウントされる。

10

【0077】

ドメインIDの欄には、対応するスレッドに割り当てられた論理ドメインの識別番号が登録される。また、対応するスレッドに論理ドメインが割り当てられていない場合、ドメインIDの欄には「none」が登録される。

【0078】

ドメイン管理部140は、管理端末などからの要求に応じて論理ドメインを設定する際や、スレッドの異常が検知された場合などに、ドメイン管理テーブル152におけるステータスおよびドメインIDの各欄を更新する。

【0079】

なお、以上のコア管理テーブル151およびドメイン管理テーブル152は、例えば、前述したハイパーバイザプログラムおよびドメイン管理プログラムが実行されるCPUチップが搭載されたビルディングブロック内のHDDに格納され、同じビルディングブロック内のメモリに展開される。

20

【0080】

次に、異常が検知されたスレッドを他のスレッドに置き換える際のサーバ装置100の処理について、フローチャートを用いて説明する。図9および図10は、スレッドの置き換え処理の例を示すフローチャートである。

【0081】

[ステップS11] 異常検知部131は、論理ドメインが割り当てられているいずれかのスレッドから異常を検知すると、その旨をドメイン管理部140に通知する。これにより、ステップS12以降の処理が開始される。

30

【0082】

[ステップS12] サーバ装置100には、コア単位での置き換えを行うか否かを示す情報があらかじめ設定されている。この情報は、ユーザの操作によって、例えば管理端末からサーバ装置100に入力される。この情報は、例えば、前述のドメイン管理プログラムと同じ不揮発性記憶装置に格納される。

【0083】

スレッドの異常が検知されたことの通知を受けたドメイン管理部140は、設定された上記の情報に基づき、コア単位で置き換えを行うか否かを判定する。コア単位で置き換えを行う場合にはステップS13の処理が実行され、行わない場合には図10のステップS21の処理が実行される。

40

【0084】

[ステップS13] ドメイン管理部140は、コア管理テーブル151に基づき、ライセンスなしのコアが存在するかを判定する。ライセンスなしのコアが存在する場合にはステップS14の処理が実行され、存在しない場合には図10のステップS21の処理が実行される。

【0085】

[ステップS14] ドメイン管理部140は、ステップS11で異常が検知されたスレッドの切り離し処理を行う。後述するように、ドメイン管理部140は、異常が検知されたスレッドに割り当てられていた論理ドメインで動作するゲストOSに対して、そのスレ

50

ッドの切り離しを指示する。また、ドメイン管理部 140 は、ドメイン管理テーブル 152 において、異常が検知されたスレッドに対応するステータスを「in use」から「fail」に更新するとともに、そのスレッドに対応するドメイン ID の欄を「none」に更新する。

【0086】

【ステップ S15】ドメイン管理部 140 は、ドメイン管理テーブル 152 に基づき、2つのスレッドのどちらにも論理ドメインが割り当てられていないコアを選択する。ドメイン管理部 140 は、選択したコアからスレッドを 1つ選択し、選択したスレッドの組み込み処理を行う。

【0087】

ドメイン管理部 140 は、ドメイン管理テーブル 152 において、選択したスレッドに対応するステータスを「free」から「in use」に更新するとともに、そのスレッドに対応するドメイン ID の欄に、異常が検知されたスレッドに割り当てられていた論理ドメインの ID を登録する。また、後述するように、ドメイン管理部 140 は、異常が検知されたスレッドに割り当てられていた論理ドメインで動作するゲスト OS に対して、選択したスレッドの組み込みを指示する。

【0088】

【ステップ S16】ドメイン管理部 140 は、ドメイン管理テーブル 152 に基づき、異常が検知されたスレッドを含むコア内の他方のスレッドに、論理ドメインが割り当てられているかを判定する。他方のスレッドに論理ドメインが割り当てられている場合にはステップ S17 の処理が実行され、割り当てられていない場合にはステップ S19 の処理が

【0089】

【ステップ S17】ドメイン管理部 140 は、他方のスレッドの切り離し処理を行う。基本的な処理手順はステップ S14 と同様であり、ドメイン管理部 140 は、他方のスレッドに割り当てられていた論理ドメインで動作するゲスト OS に対して、そのスレッドの切り離しを指示する。また、ドメイン管理部 140 は、ドメイン管理テーブル 152 において、他方のスレッドに対応するステータスを「in use」から「free」に更新するとともに、そのスレッドに対応するドメイン ID の欄を「none」に更新する。

【0090】

【ステップ S18】ドメイン管理部 140 は、ステップ S15 で選択したコア内のスレッドのうち、ステップ S15 で選択していない他方のスレッド（以下、「非選択のスレッド」と呼ぶ）の組み込み処理を行う。基本的な処理手順はステップ S15 と同様であり、ドメイン管理部 140 は、ドメイン管理テーブル 152 において、非選択のスレッドに対応するステータスを「free」から「in use」に更新するとともに、そのスレッドに対応するドメイン ID の欄に、ステップ S17 で切り離したスレッドに割り当てられていた論理ドメインの ID を登録する。また、後述するように、ドメイン管理部 140 は、ステップ S17 で切り離したスレッドに割り当てられていた論理ドメインで動作するゲスト OS に対して、非選択のスレッドの組み込みを指示する。

【0091】

【ステップ S19】ドメイン管理部 140 は、コア管理テーブル 151 における異常コア数に「1」を加算するとともに、ライセンスなしのコア数から「1」を減算する。これによりコアの置き換え処理が完了し、各論理ドメインの通常の運用が続行される。

【0092】

以上のステップ S14 ～ S19 の処理によれば、異常が検知されたスレッドを含むコアが、他のコアに置き換えられる。この場合、動作中のコアの数は変化しないため、ライセンス超過が発生することなく、異常が検知されたスレッドを置き換えることができる。

【0093】

【ステップ S21】サーバ装置 100 には、スレッド単位での置き換えを行うか否かを示す情報があらかじめ設定されている。この情報は、ユーザの操作によって、例えば管理端末からサーバ装置 100 に入力される。この情報は、例えば、前述のドメイン管理プロ

10

20

30

40

50

グラムと同じ不揮発性記憶装置に格納される。

【0094】

ドメイン管理部140は、設定された上記の情報に基づき、スレッド単位で置き換えを行うか否かを判定する。スレッド単位で置き換えを行う場合にはステップS22の処理が実行され、行わない場合にはステップS25の処理が実行される。

【0095】

〔ステップS22〕ドメイン管理部140は、ドメイン管理テーブル152に基づき、ステータスが「free」であるスレッド、すなわち、論理ドメインが未割り当てであり、かつ異常が検知されていないスレッドが存在するかを判定する。ステータスが「free」のコアが存在する場合にはステップS23の処理が実行され、存在しない場合にはステップS25の処理が実行される。

10

【0096】

〔ステップS23〕ドメイン管理部140は、ステップS11で異常が検知されたスレッドの切り離し処理を行う。この処理はステップS14と同じであるので、説明を省略する。

【0097】

〔ステップS24〕ドメイン管理部140は、ドメイン管理テーブル152に基づき、ステータスが「free」であるスレッドを選択し、選択したスレッドの組み込み処理を行う。

【0098】

基本的な処理手順はステップS15と同様であり、ドメイン管理部140は、ドメイン管理テーブル152において、選択したスレッドに対応するステータスを「free」から「in use」に更新するとともに、そのスレッドに対応するドメインIDの欄に、異常が検知されたスレッドに割り当てられていた論理ドメインのIDを登録する。また、ドメイン管理部140は、異常が検知されたスレッドに割り当てられていた論理ドメインで動作するゲストOSに対して、選択したスレッドの組み込みを指示する。

20

【0099】

これによりスレッド置き換え処理が完了し、各論理ドメインの通常の運用が続行される。

以上のステップS23、S24の処理によれば、異常が検知されたスレッドが、すでに一方のスレッドに論理ドメインが割り当てられていたコアにおける他方のスレッドに置き換えられる。この場合、動作中のコアの数は増加しないため、ライセンス超過が発生することなく、異常が検知されたスレッドを置き換えることができる。

30

【0100】

〔ステップS25〕ドメイン管理部140は、ステップS11で異常が検知されたスレッドの切り離し処理を行う。この処理はステップS14、S23と同じであるので、説明を省略する。

【0101】

図11は、スレッド切り離し処理の例を示すフローチャートである。この図11の処理は、図9のステップS14、S17、図10のステップS23、S25に対応する。

40

〔ステップS31〕ドメイン管理部140は、切り離し対象のスレッドのIDを決定する。

【0102】

〔ステップS32〕ドメイン管理部140は、切り離し対象のスレッドに割り当てられていた論理ドメインで動作するゲストOSに対して、切り離し対象のスレッドのIDを通知して、スレッドの切り離しを指示する。

【0103】

〔ステップS33〕切り離し指示を受けたゲストOSは、切り離し対象のスレッドを用いて実行されている処理を、自身に割り当てられている他のスレッドに引き継がせる。また、ゲストOSは、切り離し対象のスレッドを、自身が使用しない状態に移行させる。以

50

上の処理が完了すると、ゲストOSは、ハイパーバイザ130に対して、切り離し対象のスレッドの停止を依頼する。

【0104】

【ステップS34】ドメイン管理部140は、切り離し対象のスレッドの停止依頼を、ハイパーバイザ130を介して受け付ける。ドメイン管理部140は、ドメイン管理テーブル152において、切り離し対象のスレッドに対応するステータスを「in use」から「fail」に更新するとともに、そのスレッドに対応するドメインIDの欄を「none」に更新する。

【0105】

図12は、スレッド組み込み処理の例を示すフローチャートである。この図12の処理は、図9のステップS15、S18、図10のステップS24に対応する。

10

【ステップS41】ドメイン管理部140は、組み込み対象のスレッドのIDを決定する。

【0106】

【ステップS42】ドメイン管理部140は、ドメイン管理テーブル152において、組み込み対象のスレッドに対応するステータスを「free」から「in use」に更新するとともに、そのスレッドに対応するドメインIDの欄に、置き換え前のスレッドに割り当てられていた論理ドメインのIDを登録する。

【0107】

【ステップS43】ドメイン管理部140は、組み込み対象のスレッドに新たに割り当てた論理ドメインで動作するゲストOSに対して、組み込み対象のスレッドのIDを通知して、スレッドの組み込みを指示する。

20

【0108】

【ステップS44】組み込み指示を受けたゲストOSは、ハイパーバイザ130に対して、組み込み対象のスレッドの動作開始を依頼する。また、ゲストOSは、組み込み対象のスレッドのレジスタ、TLB(Translation Lookaside Buffer)などの初期化処理を行う。

【0109】

【ステップS45】ゲストOSは、組み込み対象のスレッドに処理を割り当てる。

次に、コア単位の置き換えを行った場合における管理テーブルの遷移例を挙げる。図13は、コア管理テーブルの遷移の例を示す図である。また、図14は、ドメイン管理テーブルの遷移の例を示す図である。

30

【0110】

図13および図14の例では、サーバ装置100は、それぞれ2つのスレッドを含む8つのコアを備えるものとする。そして、コア6台分のライセンスが設定されたものとする。初期状態では、図13の上段に示すように、コア管理テーブル151における異常コア数は「0」であり、ライセンスなしのコア数は「2」である。

【0111】

また、図14に示すように、スレッドに異常が発生する前の状態では、スレッド#0～#4に対して論理ドメインが割り当てられ、その他のスレッドには論理ドメインが割り当てられていないものとする。この場合、コア#0～#2が動作状態となるので、コア管理テーブル151においては、ドメイン割り当て済みのコア数が「3」となり、ドメイン未割り当てのコア数も「3」となる。

40

【0112】

この状態から、例としてスレッド#2から異常が検知されたものとする。コア単位での置き換えを行う場合、ドメイン管理部140は、ドメイン管理テーブル152において、スレッド#2に対応するステータスを「in use」から「fail」に更新する。また、ドメイン管理部140は、スレッド#3に対応するステータスを「in use」から「free」に更新する。さらに、ドメイン管理部140は、スレッド#2、#3にそれぞれ対応するドメインIDの欄を、ともに「1」から「none」に更新する。これにより、論理ドメイン#1が

50

らスレッド # 2 , # 3 が切り離され、スレッド # 2 , # 3 が含まれるコア # 1 の動作が禁止される。

【 0 1 1 3 】

また、ドメイン管理部 1 4 0 は、2つのスレッドがともに「free」であるコアとして、例えばコア # 3 を選択する。この場合、ドメイン管理部 1 4 0 は、コア # 3 に含まれるスレッド # 6 , # 7 に対応するステータスを、ともに「free」から「in use」に更新する。また、ドメイン管理部 1 4 0 は、スレッド # 6 に対応するドメイン ID の欄に、スレッド # 2 に割り当てられていた論理ドメインを示すドメイン ID 「 1 」を登録する。さらに、ドメイン管理部 1 4 0 は、スレッド # 7 に対応するドメイン ID の欄に、スレッド # 3 に割り当てられていた論理ドメインを示すドメイン ID 「 1 」を登録する。これにより、スレッド # 6 , # 7 が論理ドメイン # 1 に組み込まれる。

10

【 0 1 1 4 】

このとき、ドメイン管理部 1 4 0 は、コア管理テーブル 1 5 1 における異常コア数に「 1 」を加算するとともに、ライセンスなしのコア数から「 1 」を減算する。この場合、ライセンスありのコア数、すなわち、論理ドメインが割り当て済みのコア数と論理ドメインが未割り当てのコア数との合計は、「 6 」のまま変化しない。すなわち、ライセンス超過を発生させることなく、異常スレッドを他の正常なスレッドに置き換えることが可能になる。

【 0 1 1 5 】

〔 第 3 の実施の形態 〕

20

図 1 5 は、第 3 の実施の形態に係る情報処理装置が備える機能の構成例を示すブロック図である。

【 0 1 1 6 】

図 1 5 に示すサーバ装置 1 0 0 a は、第 2 の実施の形態に係るサーバ装置 1 0 0 を、仮想マシンを構築する代わりに、物理パーティションを設定するように変形したものである。図 1 5 に示す物理パーティション # 0 ~ # 3 には、サーバ装置 1 0 0 a が備えるハードウェアリソースの一部がそれぞれ割り当てられる。ここで、各物理パーティションに割り当てられる CPU は、コア内のスレッドに対応する。また、各物理パーティションには、複数の CPU、すなわち複数のスレッドが割り当てられてもよい。

【 0 1 1 7 】

30

そして、各物理パーティションにおいては、割り当てられた 1 つ以上の CPU、すなわち 1 つ以上のスレッドが、OS プログラムを実行する。これにより、各物理パーティションにおいてそれぞれ個別の OS を動作させることができる。図 1 5 の例では、物理パーティション # 0 , # 1 , # 2 , # 3 において、それぞれ OS # 0 , # 1 , # 2 , # 3 が動作する。

【 0 1 1 8 】

サーバ装置 1 0 0 a は、システム管理部 1 5 0 を備える。システム管理部 1 5 0 の処理は、第 2 の実施の形態におけるドメイン管理部 1 4 0 の処理に対応する。なお、システム管理部 1 5 0 の処理は、サーバ装置 1 0 0 a が備えるプロセッサがシステム管理プログラムを実行することで実現される。システム管理プログラムは、例えば、いずれかのビルディングブロックに搭載された少なくとも 1 つの CPU チップによって実行される。

40

【 0 1 1 9 】

また、サーバ装置 1 0 0 a の記憶装置には、コア管理テーブル 1 5 1 およびパーティション管理テーブル 1 5 2 a が記憶される。コア管理テーブル 1 5 1 に登録される情報は、第 2 の実施の形態と同様である。パーティション管理テーブル 1 5 2 a の構成は、図 8 のドメイン管理テーブル 1 5 2 における「ドメイン ID」を「パーティション ID」に置き換えた構成となる。すなわち、パーティション管理テーブル 1 5 2 a において、それぞれ個別の OS が動作するパーティションに対するスレッドの割り当てが管理される。

【 0 1 2 0 】

サービスプロセッサ 1 2 0 の処理は、サービスプロセッサ 1 2 0 の CPU 1 2 1 (図 3

50

参照)が所定のファームウェアプログラムを実行することで実現される。サービスプロセッサ120は、異常検知部131aを備える。異常検知部131aは、第2の実施の形態の異常検知部131と同様に、コア内のスレッドごとに個別のハードウェアの異常発生を検知することができる

以上のサーバ装置100aにおいては、第2の実施の形態におけるハイパーバイザ130、ドメイン管理部140およびゲストOSとの間の処理が、サービスプロセッサ120、システム管理部150およびOSとの間で実行される。すなわち、第3の実施の形態では、図9～図12に示した処理が、ハイパーバイザ130をサービスプロセッサ120に、ドメイン管理部140をシステム管理部150に、ゲストOSをOSにそれぞれ置き換えることで実現される。これにより、第3の実施の形態でも、スレッドに異常が発生した場合に、ライセンス超過を発生させることなく、異常が発生したスレッドを他の正常なスレッドに置き換えることができる。

10

【0121】

〔他のハードウェア資源の置き換え処理例〕

上記の第2,第3の実施の形態では、コアに対してライセンスを設定した場合について説明したが、例えば、サーバ装置が備える他のハードウェア資源に対してライセンスが設定されてもよい。

【0122】

ここでは、サーバ装置に搭載されたメモリ装置に対してライセンスが設定される場合について説明する。サーバ装置は、複数のメモリ装置を有しており、メモリ装置を単位としてライセンスが設定される。また、例えば、第2の実施の形態における論理ドメインごと、あるいは、第3の実施の形態における物理パーティションごとに、各メモリ装置の記憶領域を分割した分割領域を単位としてメモリ資源が割り当てられるものとする。そして、第2の実施の形態における異常検知部131、あるいは第3の実施の形態における異常検知部131aは、上記の分割領域ごとに異常を検知できるものとする。

20

【0123】

図16は、メモリ装置に対するライセンス設定の第1の例を示す図である。図16に示すビルディングブロック110には、複数のメモリモジュール310が搭載されている。メモリモジュール310は、例えば、DIMM(Dual Inline Memory Module)規格などに準拠した基板である。また、メモリモジュール310のそれぞれには、複数のメモリチップ320が搭載されている。

30

【0124】

なお、ビルディングブロックは、図2に示したビルディングブロック110a～110dのそれぞれに対応する。また、1つまたは複数のメモリモジュール310は、図2に示したメモリ112a,112bのいずれかに対応する。

【0125】

図16において、例えば、メモリモジュール310が上記のメモリ装置に対応し、メモリモジュール310内の各メモリチップ320が、上記の分割領域に対応する。すなわち、サーバ装置は、論理ドメインまたは物理パーティションに対して、メモリチップ320を単位としてメモリ資源を割り当てる。その一方で、メモリモジュール310を単位としてライセンスが設定される。

40

【0126】

サーバ装置は、メモリチップ320ごとに異常を監視する。そして、サーバ装置は、論理ドメインまたは物理パーティションに割り当て済みのいずれかのメモリチップ320から異常を検知すると、異常が検知されたメモリチップ320を含むメモリモジュール310の使用を禁止する。これとともに、サーバ装置は、使用を禁止したメモリモジュール310を、使用されていない他のメモリモジュール310(すなわち、内部のすべてのメモリチップ320が論理ドメインまたは物理パーティションに割り当てられていないメモリモジュール310)に置き換え、置き換えたメモリモジュール310の使用を開始する。

【0127】

50

このような処理により、メモリチップ 320 の異常が発生した場合でも、ライセンス超過を起こすことなく、メモリチップ 320 を置き換えることができる。

なお、図 16 の例では、メモリモジュール 310 を単位としてライセンスを設定したが、他の例として、同一メモリモジュール 310 内の隣接する複数個のメモリチップ 310 を単位としてライセンスが設定されてもよい。この場合、隣接する複数個のメモリチップ 310 の合計記憶容量を単位としてライセンスが設定されてもよい。

【0128】

図 17 は、メモリ装置に対するライセンス設定の第 2 の例を示す図である。図 17 の例は、上記のメモリ装置をメモリモジュール 310 内のメモリチップ 320 とし、上記の分割領域を、メモリチップ 320 のメモリ領域を例えば均等の容量で分割した部分領域 330 10

【0129】

図 17 の例では、サーバ装置は、論理ドメインまたは物理パーティションに対して、部分領域 330 を単位としてメモリ資源を割り当てる。その一方で、メモリチップ 320 を単位としてライセンスが設定される。

【0130】

サーバ装置は、論理ドメインまたは物理パーティションに割り当て済みのいずれかの部分領域 330 から異常を検知すると、異常が検知された部分領域 330 を含むメモリチップ 320 の使用を禁止する。これとともに、サーバ装置は、使用を禁止したメモリチップ 320 を、使用されていない他のメモリチップ 320 に置き換え、置き換えたメモリチップ 320 の使用を開始する。 20

【0131】

このような処理により、部分領域 330 の異常が発生した場合でも、ライセンス超過を起こすことなく、部分領域 330 を置き換えることができる。

以上の各実施の形態に関し、さらに以下の付記を開示する。

【0132】

(付記 1) 複数の処理を並列に実行可能であって、並列に実行する処理ごとに個別に設けられた個別回路を有する演算回路を、複数備えた情報処理装置において、

前記演算回路を単位として設定された使用権の数を記憶する記憶部と、

複数の前記演算回路のうち、前記使用権の数の前記演算回路の動作を許可するとともに、動作中の前記演算回路に搭載されたいずれかの前記個別回路の異常を検知すると、異常が検知された前記個別回路が搭載された前記演算回路における他の前記個別回路の動作を停止させ、動作していないいずれかの前記演算回路の動作を開始させる制御部と、 30

を有することを特徴とする情報処理装置。

【0133】

(付記 2) 前記制御部は、動作中の前記演算回路に搭載されたいずれかの前記個別回路の異常を検知したとき、動作していない前記演算回路が存在しない場合には、動作中の前記演算回路の中に使用されていない前記個別回路があるかを判定し、使用されていない前記個別回路がある場合には、異常が検知された前記個別回路を用いて実行されていた処理を、使用されていない前記個別回路を用いて実行させることを特徴とする付記 1 記載の情報処理装置。 40

【0134】

(付記 3) 前記記憶部は、動作中の前記演算回路に搭載された前記個別回路のうち、1 つまたは複数のオペレーティングシステムのいずれかの処理に割り当てられた前記個別回路の情報を保持する管理情報を記憶し、

前記制御部は、前記管理情報においていずれかのオペレーティングシステムの処理に割り当てられた前記個別回路の異常を検知したとき、動作していない前記演算回路が存在しない場合には、前記管理情報に基づき、動作中の前記演算回路に搭載された前記個別回路の中にいずれかのオペレーティングシステムの処理にも割り当てられていない前記個別回路があるかを判定し、いずれかのオペレーティングシステムの処理にも割り当てられていない 50

前記個別回路がある場合には、異常が検知された前記個別回路がいずれのオペレーティングシステムにも割り当てられていない前記個別回路に置き換えられるように前記管理情報を更新する、

ことを特徴とする付記 1 記載の情報処理装置。

【0135】

(付記 4) 前記情報処理装置は、複数のメモリ装置を有し、

前記記憶部は、前記メモリ装置を単位として設定されたメモリ使用権の数を記憶し、

前記制御部は、複数の前記メモリ装置のうち、前記メモリ使用権の数の前記メモリ装置の使用を許可するとともに、使用中の前記メモリ装置の記憶領域を分割した分割領域のいずれかから異常を検知すると、異常が検知された前記分割領域を含む前記メモリ装置における他の前記分割領域の使用を停止させ、使用されていないいずれかの前記メモリ装置の使用を開始する、

10

ことを特徴とする付記 1 ~ 3 のいずれか 1 つに記載の情報処理装置。

【0136】

(付記 5) 複数の処理を並列に実行可能であって、並列に実行する処理ごとに個別に設けられた個別回路を有する演算回路を、複数備えた情報処理装置における制御方法であって、

前記演算回路を単位として設定された使用権の数を記憶する記憶部を参照して、複数の前記演算回路のうち、前記使用権の数の前記演算回路の動作を許可し、

動作中の前記演算回路に搭載されたいずれかの前記個別回路の異常を検知すると、異常が検知された前記個別回路が搭載された前記演算回路における他の前記個別回路の動作を停止させ、動作していないいずれかの前記演算回路の動作を開始させる、

20

ことを特徴とする制御方法。

【0137】

(付記 6) 動作中の前記演算回路に搭載されたいずれかの前記個別回路の異常を検知したとき、動作していない前記演算回路が存在しない場合には、動作中の前記演算回路の中に使用されていない前記個別回路があるかを判定し、使用されていない前記個別回路がある場合には、異常が検知された前記個別回路を用いて実行されていた処理を、使用されていない前記個別回路を用いて実行させる、

処理をさらに含むことを特徴とする付記 5 記載の制御方法。

30

【0138】

(付記 7) 前記記憶部は、動作中の前記演算回路に搭載された前記個別回路のうち、1 つまたは複数のオペレーティングシステムのいずれかの処理に割り当てられた前記個別回路の情報を保持する管理情報を記憶し、

前記管理情報においていずれかのオペレーティングシステムの処理に割り当てられた前記個別回路の異常を検知したとき、動作していない前記演算回路が存在しない場合には、前記管理情報に基づき、動作中の前記演算回路に搭載された前記個別回路の中にいずれのオペレーティングシステムの処理にも割り当てられていない前記個別回路があるかを判定し、いずれのオペレーティングシステムの処理にも割り当てられていない前記個別回路がある場合には、異常が検知された前記個別回路がいずれのオペレーティングシステムにも割り当てられていない前記個別回路に置き換えられるように前記管理情報を更新する、

40

処理をさらに含むことを特徴とする付記 5 記載の制御方法。

【0139】

(付記 8) 前記情報処理装置は、複数のメモリ装置を有し、

前記記憶部は、前記メモリ装置を単位として設定されたメモリ使用権の数を記憶し、

複数の前記メモリ装置のうち、前記メモリ使用権の数の前記メモリ装置の使用を許可し、使用中の前記メモリ装置の記憶領域を分割した分割領域のいずれかから異常を検知すると、異常が検知された前記分割領域を含む前記メモリ装置における他の前記分割領域の使用を停止させ、使用されていないいずれかの前記メモリ装置の使用を開始する、

処理をさらに含むことを特徴とする付記 5 ~ 7 のいずれか 1 つに記載の制御方法。

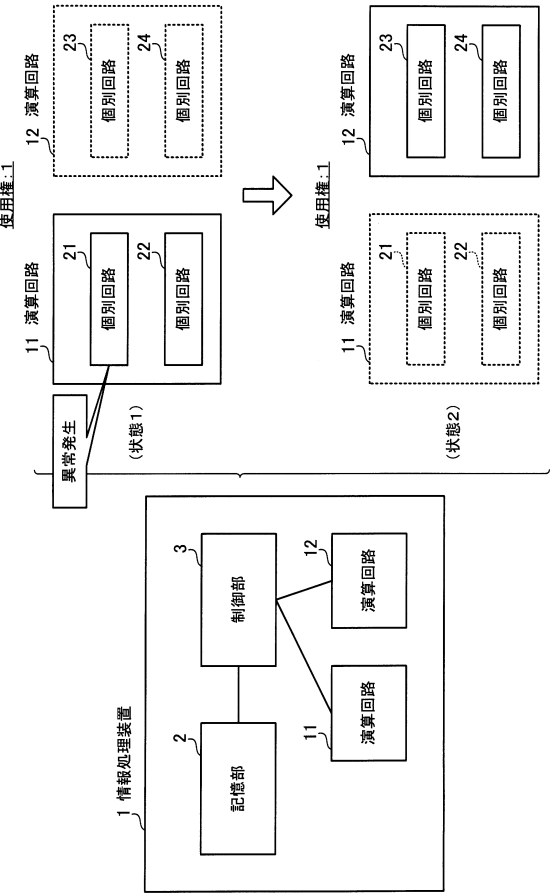
50

【符号の説明】

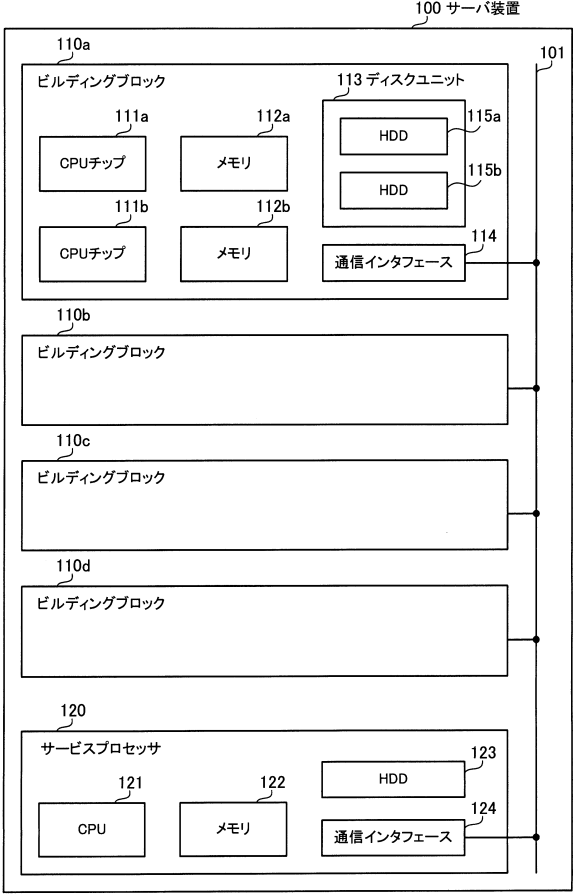
【 0 1 4 0 】

- 1 情報処理装置
- 2 記憶部
- 3 制御部
- 11, 12 演算回路
- 21 ~ 24 個別回路

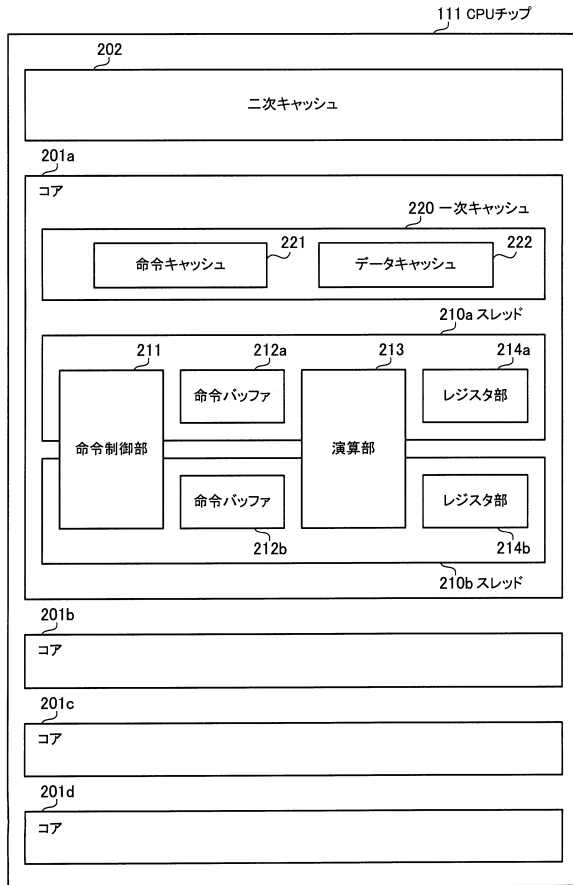
【図 1】



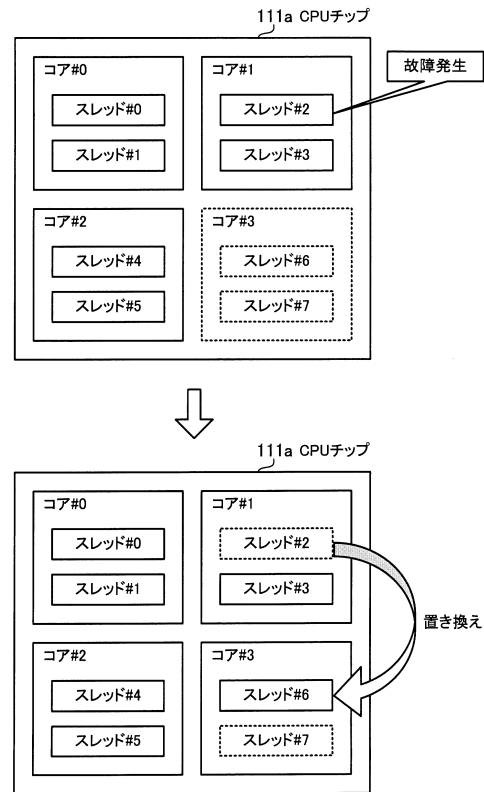
【図 2】



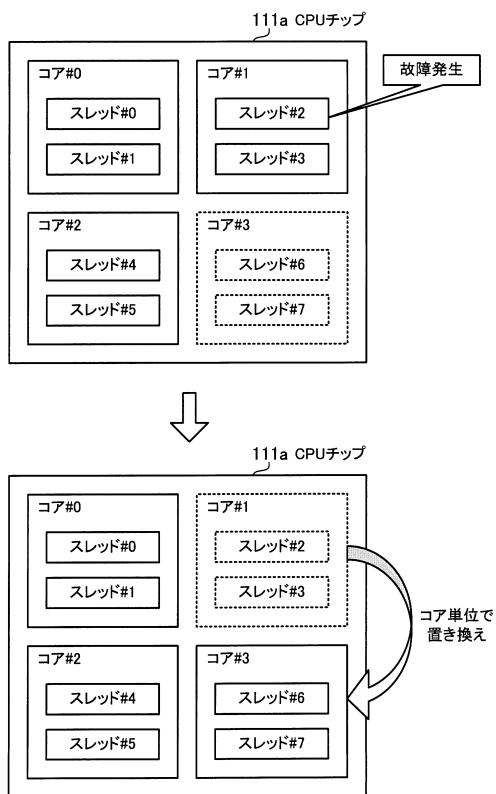
【 図 3 】



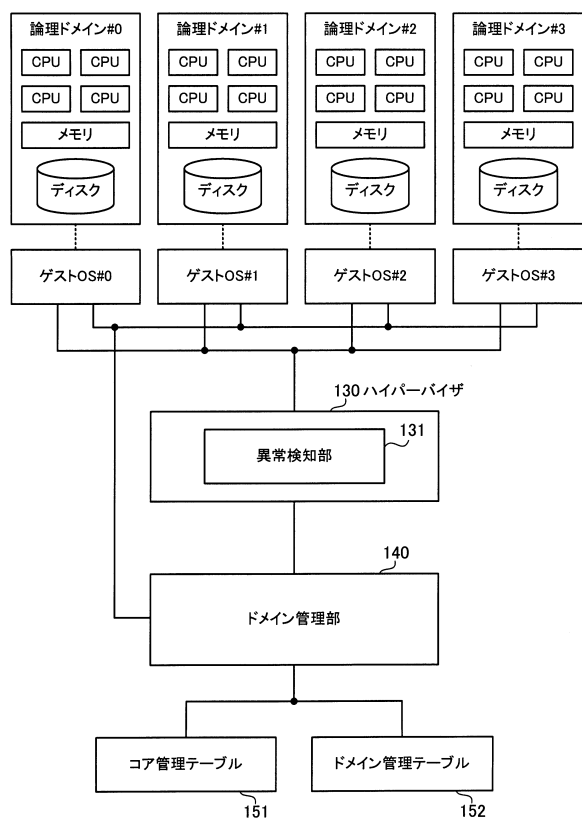
【 図 4 】



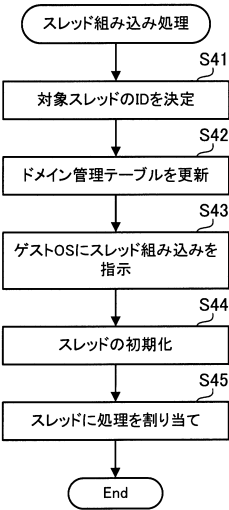
【 図 5 】



【圖 6】



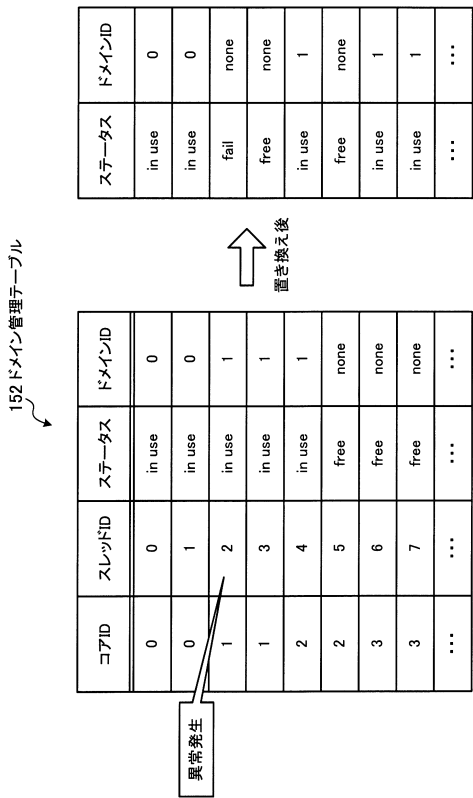
【図 1 2】



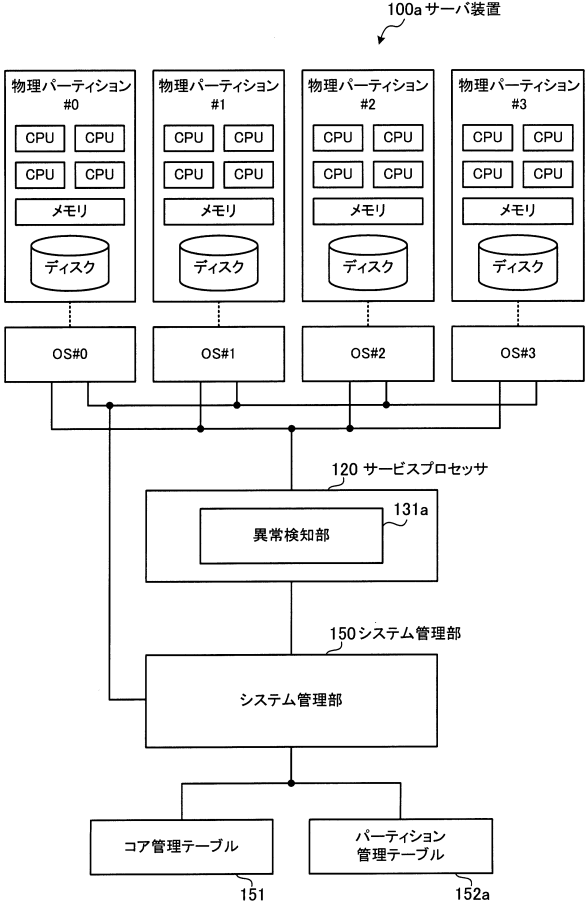
【図 1 3】



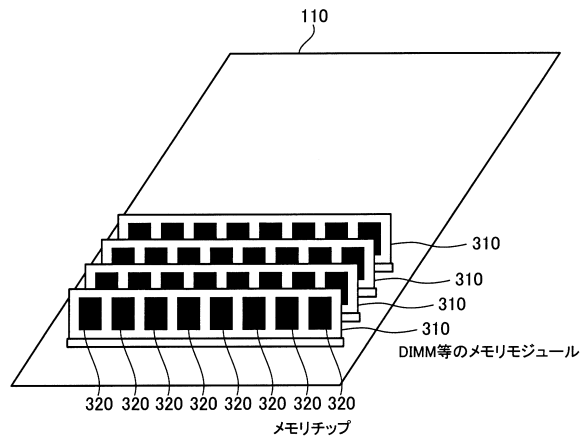
【図 1 4】



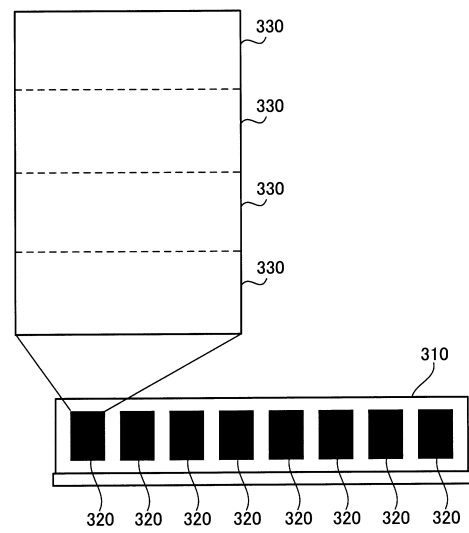
【図 1 5】



【図 16】



【図 17】



フロントページの続き

- (56)参考文献 特開2009-205696(JP,A)
国際公開第2010/046966(WO,A1)
特開2011-227840(JP,A)
日本アイ・ビー・エム株式会社,「AIX 論理分割(LPAR)とAIX 5L V5.2」
 , 株式会社アスキー 小森 哲郎,2003年11月25日,第1版,pp.23-36
- (58)調査した分野(Int.Cl.,DB名)
G06F 11/20
G06F 9/46 - 9/54