

(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2012-164032

(P2012-164032A)

(43) 公開日 平成24年8月30日(2012.8.30)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 1/32 (2006.01)	G06F 1/00 332Z	5B011
G06F 9/46 (2006.01)	G06F 9/46 350	5B079
G06F 1/04 (2006.01)	G06F 1/04 301C	

審査請求 未請求 請求項の数 5 O L (全 21 頁)

(21) 出願番号 特願2011-22149 (P2011-22149)
 (22) 出願日 平成23年2月3日(2011.2.3)

(71) 出願人 000005223
 富士通株式会社
 神奈川県川崎市中原区上小田中4丁目1番1号
 (74) 代理人 100104190
 弁理士 酒井 昭徳
 (72) 発明者 村上 岳生
 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 野田 政秀
 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 矢崎 昌朋
 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

最終頁に続く

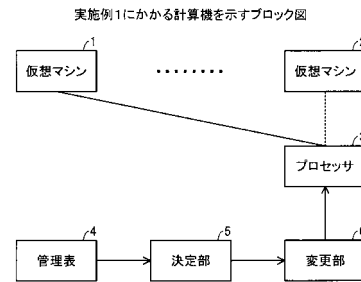
(54) 【発明の名称】 計算機、制御方法及びプログラム

(57) 【要約】

【課題】 計算機の消費電力を削減すること。

【解決手段】 計算機は、1個以上の物理的なプロセッサ3、管理表4、決定部5及び変更部6を備えている。管理表4は、複数の仮想マシン1, 2ごとに、クロック周波数を制御する制御情報を管理する。仮想マシン1, 2は、物理的なプロセッサ3で実行されることによって実現される。制御情報は、仮想マシン1, 2の種別に応じて設定されている。決定部5は、仮想マシン1, 2ごとに、管理表4の制御情報に基づいてしきい値を求める。決定部5は、仮想マシン1, 2ごとに、しきい値と物理的なプロセッサ3の利用率とに基づいてクロック周波数を決定する。変更部6は、決定部5により決定されたクロック周波数に基づいて物理的なプロセッサ3のクロック周波数を変更する。

【選択図】 図1



【特許請求の範囲】

【請求項 1】

1 個以上の物理的なプロセッサと、
前記物理的なプロセッサで実行されることによって実現される複数の仮想マシンごとに、前記仮想マシンの種別に応じて設定される、クロック周波数を制御する制御情報を管理する管理表と、

前記仮想マシンごとに、前記管理表の制御情報に基づいてしきい値を求め、前記しきい値と前記物理的なプロセッサの利用率とに基づいてクロック周波数を決定する決定部と、

前記決定部により決定されたクロック周波数に基づいて前記物理的なプロセッサのクロック周波数を変更する変更部と、

を備えることを特徴とする計算機。

10

【請求項 2】

前記制御情報は、前記仮想マシンごとに、前記仮想マシンがフォアグラウンドで動作しているか、バックグラウンドで動作しているか、に応じて設定される情報を含むことを特徴とする請求項 1 に記載の計算機。

【請求項 3】

前記決定部は、前記仮想マシンごとに、前記しきい値の最大値と最小値とを求め、前記物理的なプロセッサの利用率が前記しきい値の最大値よりも大きい場合にクロック周波数を上げ、前記物理的なプロセッサの利用率が前記しきい値の最小値よりも小さい場合にクロック周波数を下げるように決定することを特徴とする請求項 1 または 2 に記載の計算機。

20

【請求項 4】

物理的なプロセッサの利用率を求め、

前記物理的なプロセッサで実行されることによって実現される複数の仮想マシンごとに、前記仮想マシンの種別に応じて設定される、クロック周波数を制御する制御情報に基づいてしきい値を求め、

前記仮想マシンごとに、前記しきい値と前記物理的なプロセッサの利用率とに基づいてクロック周波数を決定することを特徴とする制御方法。

【請求項 5】

物理的なプロセッサの利用率を求め、

前記物理的なプロセッサで実行されることによって実現される複数の仮想マシンごとに、前記仮想マシンの種別に応じて設定される、クロック周波数を制御する制御情報をメモリから読み出して前記制御情報に基づいてしきい値を求め、

前記仮想マシンごとに、前記しきい値と前記物理的なプロセッサの利用率とに基づいてクロック周波数を決定する、

処理をコンピュータに実行させるプログラム。

30

【発明の詳細な説明】

【技術分野】

【0001】

この発明は、計算機、制御方法及びプログラムに関する。

40

【背景技術】

【0002】

従来、計算機において、一つの物理的な CPU (Central Processing Unit、中央処理装置) 上で複数の OS (Operating System、オペレーティングシステム) を動作させて複数の仮想マシンを実行する仮想化システムがある。仮想化システムにおいて、各仮想マシンへ仮想バッテリーを提供し、仮想マシンが駆動中であれば CPU 使用時間を放電量に換算し、仮想マシンが停止中であれば停止時間を充電量に換算して、仮想バッテリーの残量を制御する技術がある。また、物理的な CPU に割り当てられている仮想マシンを削除した結果、仮想マシンの割り当てがなくなった物理的な CPU をスリープ状態に移行させることによって、消費電力の低減を図る技術がある。

50

また、データ処理システム内に生成された区画に、物理リソースのセットのうちの少なくともいくつかの物理リソースを割り当て、割り当てられなかった物理リソースの電力消費を低減する技術がある。また、物理リソースに供給するクロック周波数を低くすることにより、電力消費量を低減する技術がある。

【先行技術文献】

【特許文献】

【0003】

【特許文献1】特開2010-33207号公報

【特許文献2】特開2009-140157号公報

【特許文献3】特開2004-192612号公報

10

【発明の概要】

【発明が解決しようとする課題】

【0004】

しかしながら、従来の技術では、物理的なCPUの負荷の状態にかかわらず、各仮想マシンは、予め設定されているクロック周波数で動作する。そのため、物理的なCPUが負荷の低い状態にあるため、各仮想マシンを低速で実行することができる状態であるにもかかわらず、仮想マシンに高いクロック周波数が設定されている場合には、物理的なCPUが高いクロック周波数で動作することがある。また、物理的なCPUの負荷に応じて物理的なCPUのクロック周波数の上げ下げを行う場合、クロック周波数が低くてもかまわない仮想マシンが物理的なCPUに割り当てられているときでも、物理的なCPUの負荷が高い状態のときには、物理的なCPUが高いクロック周波数で動作することがある。このように、物理的なCPUが低いクロック周波数で動作することができる状況のときでも、それよりも高いクロック周波数で物理的なCPUが動作することがあるため、無駄な電力を消費してしまうという問題点がある。

20

【0005】

消費電力を削減することができる計算機、制御方法及びプログラム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】

【0006】

計算機は、1個以上の物理的なプロセッサ、管理表、決定部及び変更部を備えている。管理表は、複数の仮想マシンごとに、クロック周波数を制御する制御情報を管理する。仮想マシンは、物理的なプロセッサで実行されることによって実現される。制御情報は、仮想マシンの種別に応じて設定されている。決定部は、仮想マシンごとに、管理表の制御情報に基づいてしきい値を求める。決定部は、仮想マシンごとに、しきい値と物理的なプロセッサの利用率とに基づいてクロック周波数を決定する。変更部は、決定部により決定されたクロック周波数に基づいて物理的なプロセッサのクロック周波数を変更する。

30

【発明の効果】

【0007】

この計算機、制御方法及びプログラムによれば、消費電力を削減することができる。

【図面の簡単な説明】

40

【0008】

【図1】図1は、実施例1にかかる計算機を示すブロック図である。

【図2】図2は、実施例2にかかる計算機のハードウェア構成を示すブロック図である。

【図3】図3は、実施例2にかかる計算機の機能的な構成を示すブロック図である。

【図4】図4は、実施例2にかかる計算機の管理表の一例を示す図表である。

【図5】図5は、実施例2にかかる制御方法におけるCPU周波数定期更新処理を示すフローチャートである。

【図6】図6は、実施例2にかかる制御方法における仮想マシン割り当て処理を示すフローチャートである。

【図7】図7は、実施例2にかかる制御方法におけるCPU周波数切り替え処理を示すフ

50

ローチャートである。

【図 8】図 8 は、実施例 2 にかかる制御方法におけるフロント係数設定変更処理を示すフローチャートである。

【図 9】図 9 は、CPU の利用率が下降していく例における各種情報の一例を示す図表である。

【図 10】図 10 は、CPU の利用率が下降していく例における CPU 周波数の一例を示す模式図である。

【図 11】図 11 は、CPU の利用率が上昇していく例における各種情報の一例を示す図表である。

【図 12】図 12 は、CPU の利用率が上昇していく例における CPU 周波数の一例を示す模式図である。

【図 13】図 13 は、フォアグラウンドで動作する仮想マシンを切り替える例における各種情報の一例を示す図表である。

【図 14】図 14 は、フォアグラウンドで動作する仮想マシンを切り替える例における CPU 周波数の一例を示す模式図である。

【図 15】図 15 は、新たに仮想マシンが起動する例における各種情報の一例を示す図表である。

【図 16】図 16 は、新たに仮想マシンが起動する例における CPU 周波数の一例を示す模式図である。

【発明を実施するための形態】

【0009】

以下に添付図面を参照して、この計算機、制御方法及びプログラムの好適な実施の形態を詳細に説明する。計算機、制御方法及びプログラムは、仮想マシンごとに、仮想マシンの種別に応じた制御情報に基づいてしきい値を求め、しきい値とプロセッサの利用率（負荷）とに基づいてクロック周波数を決定し、プロセッサのクロック周波数を決定したクロック周波数に変更するものである。以下の各実施例の説明においては、同様の構成要素には同一の符号を付して、重複する説明を省略する。

【0010】

（実施例 1）

図 1 は、実施例 1 にかかる計算機を示すブロック図である。図 1 に示すように、計算機は、1 個以上の物理的なプロセッサ 3、管理表 4、決定部 5 及び変更部 6 を備えている。

【0011】

管理表 4 は、複数の仮想マシン 1, 2 ごとに、クロック周波数を制御する制御情報を管理する。仮想マシン 1, 2 は、物理的なプロセッサ 3 で実行されることによって実現される。制御情報は、仮想マシン 1, 2 の種別に応じて設定されている。決定部 5 は、仮想マシン 1, 2 ごとに、管理表 4 の制御情報に基づいてしきい値を求める。決定部 5 は、仮想マシン 1, 2 ごとに、しきい値と物理的なプロセッサ 3 の利用率とに基づいてクロック周波数を決定する。変更部 6 は、決定部 5 により決定されたクロック周波数に基づいて物理的なプロセッサ 3 のクロック周波数を変更する。

【0012】

実施例 1 によれば、仮想マシン 1, 2 の種別に応じて仮想マシン 1, 2 のクロック周波数が決まり、それに応じて物理的なプロセッサ 3 のクロック周波数を変更される。例えば、物理的なプロセッサ 3 の利用率に連動してクロック周波数が変動する仮想マシン 1, 2 の場合には、物理的なプロセッサ 3 の利用率に連動したクロック周波数で物理的なプロセッサ 3 が動作する。物理的なプロセッサ 3 の利用率に連動し、かつ重み付けされてクロック周波数が変動する仮想マシン 1, 2 の場合には、物理的なプロセッサ 3 の利用率に連動し、かつ重み付けされたクロック周波数で物理的なプロセッサ 3 が動作する。一定のクロック周波数で動作する仮想マシン 1, 2 の場合には、一定のクロック周波数で物理的なプロセッサ 3 が動作する。つまり、物理的なプロセッサ 3 が低いクロック周波数で動作することができる状況のときには、物理的なプロセッサ 3 が低いクロック周波数で動作するの

10

20

30

40

50

で、計算機の消費電力を削減することができる。

【0013】

(実施例2)

・ハードウェア構成の説明

図2は、実施例2にかかる計算機のハードウェア構成を示すブロック図である。図2に示すように、計算機は、物理的なプロセッサとして例えばCPU11、表示部12、キー13、メインメモリ14及びNAND(Not AND)メモリ15を備えている。CPU11、表示部12、キー13、メインメモリ14及びNANDメモリ15は、バス16に接続されている。

【0014】

CPU11は、計算機全体の制御を司る。CPU11は、後述するプログラムを実行し、後述する種々の仮想マシン(VM:Virtual Machine)や仮想マシンモニタを実現する。CPU11は、クロック周波数(CPU周波数)や駆動電圧を動的に変更することができるものである。表示部12は、例えば液晶パネルを備えている。キー13は、例えば数字や文字を入力するためのキーやボタンを備えている。キー13は、表示部12と一体化されたタッチパネルを備えており、表示部12に表示されたキーの画像に触れることにより数字や文字を入力することができるようになっていてもよい。

【0015】

メインメモリ14は、CPU11の作業領域として用いられる。メインメモリ14には、例えばOSやアプリケーションのプログラムが展開されてもよい。CPU11は、メインメモリ14に展開されたOSやアプリケーションのプログラムを実行してもよい。メインメモリ14は、例えばRAM(Random Access Memory)を備えている。

【0016】

NANDメモリ15は、例えばOSやアプリケーションのプログラムを格納している。OSやアプリケーションのプログラムは、NANDメモリ15から読み出されてメインメモリ14に展開されてもよい。NANDメモリ15は、不揮発性のメモリとして、例えばNAND型のフラッシュメモリを備えていてもよい。計算機は、NANDメモリ15の代わりに、他の不揮発性メモリを備えていてもよい。

【0017】

なお、計算機が携帯電話機などの無線通信端末である場合には、計算機は無線通信部を備えている。また、計算機は、画像処理や音声処理を行うDSP(Digital Signal Processor)や、外部メモリとのインタフェースなどを備えていてもよい。

【0018】

・機能的な構成の説明

図3は、実施例2にかかる計算機の機能的な構成を示すブロック図である。図3に示すように、計算機は、複数の仮想マシン21~25及び仮想マシンモニタ26を備えている。仮想マシン21~25及び仮想マシンモニタ26は、CPU11が仮想マシン及び仮想マシンモニタを実現するプログラムを実行することにより実現されてもよい。例えば各仮想マシン21~25では、独立してOSが実行されてもよい。以下の説明では、仮想マシン21~25のID(Identifier、識別子)を例えばA、B、C、D及びEとする。

【0019】

仮想マシン21,22は、ゲスト仮想マシン(以下、ゲストVMと表記する)である。ゲスト仮想マシン21及びゲスト仮想マシン22を、それぞれ、ゲストVM-A21及びゲストVM-B22と表記することがある。ゲストVM-A21及びゲストVM-B22は、ユーザーが使用するアプリケーションを動作させる仮想マシンである。ゲストVMは、使用形態に応じて、一つだけ起動されてもよいし、複数起動されてもよい。

【0020】

10

20

30

40

50

ゲストVMは、例えばフロントエンドドライバ27, 28を備えている。フロントエンドドライバ27, 28は、ゲストVMがデバイス进行操作する際に利用されるモジュールである。フロントエンドドライバ27, 28は、後述するドライバ仮想マシンと連携して、デバイス进行操作する処理を行う。

【0021】

仮想マシン23, 24は、ドライバ仮想マシン(以下、ドライバVMと表記する)である。ドライバ仮想マシン23及びドライバ仮想マシン24を、それぞれ、ドライバVM-C23及びドライバVM-D24と表記することがある。ドライバVM-C23及びドライバVM-D24は、デバイスを制御するデバイスドライバを動作させる仮想マシンである。デバイス種別ごとにドライバVMが起動されることがあるので、複数のドライバVMが起動されることがある。

10

【0022】

ドライバVMは、例えばバックエンドドライバ29, 30を備えている。バックエンドドライバ29, 30は、ゲストVMのフロントエンドドライバ27, 28からの要求を受けて、デバイスの操作を行う。図示例では、例えばドライバVM-C23のバックエンドドライバ29は、キー13の操作を行う。例えばドライバVM-D24のバックエンドドライバ30は、NANDメモリ15の操作を行う。

【0023】

仮想マシン25は、サービス仮想マシン(以下、サービスVMと表記する)である。サービス仮想マシン25をサービスVM-E25と表記することがある。サービスVM-E25は、ユーザーにより直接操作されるわけではなく、バックグラウンドで動作する。サービスVM-E25は、例えばウイルスチェックなどシステムに必要な補助的なサービスを実施する。

20

【0024】

仮想マシンモニタ26は、一つのCPU11を複数の仮想マシンとして利用する制御を行う。それによって、一つのCPU11を複数の仮想マシンとして利用することができる。仮想マシンモニタ26は、決定部として例えばゲスト電力制御モジュール31を備えている。ゲスト電力制御モジュール31は、例えば管理表としてゲスト電力制御管理表32を備えている。

【0025】

ゲスト電力制御モジュール31は、一定間隔またはゲストVMの切り替え時に動作して、ゲスト電力制御管理表32を参照し、更新の必要があるときにはゲスト電力制御管理表32を更新する。ゲスト電力制御モジュール31は、各仮想マシンのCPU周波数を変更する必要があるか否かを判断し、変更の必要がある場合には後述するCPU周波数切り替え処理を行う。

30

【0026】

ゲスト電力制御管理表32は、仮想マシンごとに、仮想マシンの種別や動作状態に応じて電力制御を行うのに必要な情報を保持する。ゲスト電力制御管理表32は、例えばメインメモリ14やNANDメモリ15に記憶される。仮想マシンの種別には、例えば上述したゲストVMやドライバVMやサービスVMなどがある。仮想マシンの動作状態には、例えばフォアグラウンドで動作している状態とバックグラウンドで動作している状態がある。

40

【0027】

また、ゲスト電力制御管理表32の一部として、あるいはゲスト電力制御管理表32とは別に、後述するMaxデフォルト値及びMinデフォルト値、並びにフロント係数のデフォルト値が、例えばメインメモリ14やNANDメモリ15に記憶されていてもよい。

【0028】

仮想マシンモニタ26は、画面切り替え制御モジュール33、CPUスケジューラ34、及び変更部として例えばCPU周波数変更処理モジュール35を備えている。画面切り替え制御モジュール33は、ユーザーが、計算機の図示しない画面切り替えキーを押下す

50

るなどの画面の切り替え操作を行ったときに呼び出される。画面切り替え制御モジュール 33 は、新規にフロント画面で操作されるゲスト VM、すなわちフォアグラウンドで動作するゲスト VM を決定し、フロント画面で操作されるゲスト VM を切り替える。

【0029】

CPUスケジューラ 34 は、CPU 11 に仮想マシンを割り当てる制御を行う。割り当て方の一例として、例えば CREDIT や SEDF (Simple Earliest Deadline First) や BVT (Borrowed Virtual Time) などのポリシーがある。また、複数のポリシーの中から適宜選択して割り当てるようにしてもよい。CPUスケジューラ 34 は、CPU 11 で動作させる仮想マシンの切り替えに合わせてゲスト電力制御モジュール 31 の CPU 周波数切り替え処理を呼び出す。CPU 周波数変更処理モジュール 35 は、CPU 11 の CPU 周波数を変更する処理を実行する。

10

【0030】

・ゲスト電力制御管理表の一例

図 4 は、実施例 2 にかかる計算機の管理表の一例を示す図表である。図 4 に示すように、ゲスト電力制御管理表 32 は、例えば VM_ID フィールド、VM タイプ係数フィールド、フロント係数フィールド及び CPU 周波数フィールドを有する。VM_ID フィールドには、仮想マシンの ID が格納される。VM タイプ係数フィールドには、仮想マシンの種別に関する制御情報として VM タイプ係数の値が格納される。仮想マシンの種別に関する制御情報は、仮想マシンの種別に応じて予め設定されていてもよい。

20

【0031】

フロント係数フィールドには、仮想マシンの動作状態に関する制御情報としてフロント係数の値が格納される。仮想マシンの動作状態に関する制御情報は、仮想マシンがフォアグラウンドで動作しているか、バックグラウンドで動作しているか、によって動的に変化してもよい。CPU 周波数フィールドには、仮想マシンが動作する際に CPU 11 に設定される CPU 周波数が格納される。

【0032】

・仮想マシンの CPU 周波数の決定方法の一例

VM タイプ係数の値及びフロント係数の値は、例えば次の (1) 式及び (2) 式において用いられる。(1) 式及び (2) 式において、しきい値 Max 及びしきい値 Min は、それぞれ、CPU 11 の利用率と比較されるしきい値の最大値及び最小値である。ゲスト電力制御モジュール 31 は、CPU 11 の利用率がしきい値 Max よりも大きい場合には仮想マシンの CPU 周波数を高くし、CPU 11 の利用率がしきい値 Min よりも小さい場合には仮想マシンの CPU 周波数を低くするように、制御する。Max デフォルト値及び Min デフォルト値は、それぞれ、しきい値の最大値及び最小値のデフォルト値であり、予め設定されていてもよい。

30

【0033】

【数 1】

$$[\text{しきい値Max}] = [\text{Maxデフォルト値}] \times [\text{VMタイプ係数値}] \times [\text{フロント係数値}] \dots(1)$$

40

【0034】

【数 2】

$$[\text{しきい値Min}] = [\text{Minデフォルト値}] \times [\text{VMタイプ係数値}] \times [\text{フロント係数値}] \dots(2)$$

【0035】

(1)式及び(2)式によれば、VMタイプ係数の値が大きいと、しきい値Max及びしきい値Minがともに大きくなるので、仮想マシンのCPU周波数は、CPU11の利用率が大きくなっても高くなりやすく、かつCPU11の利用率が小さくなると低くなりやすい。つまり、VMタイプ係数の値が大きい仮想マシンは、できるだけ低いCPU周波数で動作するように制御される。

10

【0036】

それに対して、VMタイプ係数の値が小さいと、しきい値Max及びしきい値Minがともに小さくなるので、仮想マシンのCPU周波数は、CPU11の利用率が大きくなると高くなりやすく、かつCPU11の利用率が小さくなっても低くなりやすい。つまり、VMタイプ係数の値が小さい仮想マシンは、できるだけ高いCPU周波数で動作するように制御される。

20

【0037】

例えば図4に示す例では、VMタイプ係数の標準値は例えば1である。仮想マシンのIDがA及びBであるゲストVMのVMタイプ係数は標準値に設定されている。ゲストVMは、CPU11の利用率に連動してCPU周波数が変動する仮想マシンである。

【0038】

仮想マシンのIDがC及びDであるドライバVMのVMタイプ係数は標準値や標準値と異なる値、例えば1や1.2に設定されている。ドライバVMは、CPU11の利用率に連動し、かつ重み付けされてCPU周波数が変動する仮想マシンである。例えば、ドライバVM-CのVMタイプ係数の値は1であり、ドライバVM-DのVMタイプ係数の値は1.2であるので、ドライバVM-CはドライバVM-Dよりも高いCPU周波数で動作しやすいように設定されていることになる。

30

【0039】

仮想マシンのIDがEであるサービスVMのVMタイプ係数の値は例えば0に設定されている。VMタイプ係数の値が例えば0に設定されている仮想マシンは、CPU11の利用率に連動しないで、一定のCPU周波数で動作する仮想マシンである。

【0040】

また、フロント係数の値についても同様であり、フロント係数の値が大きい仮想マシンは、できるだけ低いCPU周波数で動作するように制御される。フロント係数の値が小さい仮想マシンは、できるだけ高いCPU周波数で動作するように制御される。

【0041】

例えば図4に示す例では、フロント係数のデフォルト値は例えば0.9である。仮想マシンのIDがAであるゲストVMのフロント係数はデフォルト値に設定されている。仮想マシンのIDがBであるゲストVMのフロント係数はデフォルト値よりも大きい値、例えば1に設定されている。従って、ゲストVM-AはゲストVM-Bよりも高いCPU周波数で動作しやすいように設定されていることになり、ゲストVM-Aがフォアグラウンドで動作し、ゲストVM-Bがバックグラウンドで動作していることになる。

40

【0042】

・制御方法の説明

[CPU周波数定期更新処理]

図5は、実施例2にかかる制御方法におけるCPU周波数定期更新処理を示すフローチャートである。CPU周波数定期更新処理は、ゲスト電力制御モジュール31により一定

50

間隔で定期的実施される。

【0043】

図5に示すように、CPU周波数定期更新処理は、例えば計算機に内蔵されたタイマーの割り込みにより一定間隔おきに起動される。CPU周波数定期更新処理が開始されると、ゲスト電力制御モジュール31は、まず、例えばCPUスケジューラ34のCPU負荷計測モジュール(図3において図示省略)を呼び出し、直近の計算機全体でのCPU11の利用率を算出する(ステップS1)。

【0044】

次いで、ゲスト電力制御モジュール31は、ゲスト電力制御管理表32を参照し、一番目の仮想マシンのVMタイプ係数の値が0であるか否かを判断する(ステップS2)。なお、ステップS2をステップS1の前に行ってもよい。VMタイプ係数の値が0でない場合(ステップS2:No)、ゲスト電力制御モジュール31は、ゲスト電力制御管理表32を参照し、例えば上述した(1)式及び(2)式に基づいて、しきい値Max及びしきい値Minを計算する(ステップS3)。なお、ステップS1の前にステップS2及びステップS3を行ってもよい。

10

【0045】

次いで、ゲスト電力制御モジュール31は、ステップS1で算出したCPU11の利用率がしきい値Maxよりも大きいかが、またはしきい値Minよりも小さいかが、を判断する(ステップS4)。しきい値Maxよりも大きいか、またはしきい値Minよりも小さい場合(ステップS4:Yes)、ゲスト電力制御モジュール31は、当該仮想マシンのCPU周波数を決定する。そして、ゲスト電力制御モジュール31は、決定したCPU周波数が現在の設定値と異なる場合には、ゲスト電力制御管理表32のCPU周波数フィールドの、当該仮想マシンに対応する格納値を更新する(ステップS5)。

20

【0046】

ステップS5では、例えば、CPU11の利用率がしきい値Maxよりも大きい場合には、当該仮想マシンのCPU周波数を現在の設定値よりも高くしてもよい。CPU周波数を現在の設定値よりも高くする場合には、例えば複数段階に設定可能なCPU周波数の最も高い周波数が設定されてもよい。例えばCPU周波数が600MHz、500MHz、250MHz及び125MHzの4段階に設定可能である場合、最も高い600MHzが設定されてもよい。

30

【0047】

また、例えば、CPU11の利用率がしきい値Minよりも小さい場合には、当該仮想マシンのCPU周波数を現在の設定値よりも低くしてもよい。CPU周波数を現在の設定値よりも低くする場合には、例えば複数段階に設定可能なCPU周波数の一段低い周波数が設定されてもよい。例えばCPU周波数が600MHz、500MHz、250MHz及び125MHzの4段階に設定可能であり、現在の設定値が500MHzである場合、一段低い250MHzが設定されてもよい。

【0048】

次いで、ゲスト電力制御モジュール31は、ゲスト電力制御管理表32のCPU周波数フィールドの格納値が更新された仮想マシンが現在、CPU11で動作中である場合には、CPU周波数変更処理モジュール35を呼び出す。呼び出されたCPU周波数変更処理モジュール35は、CPU11のCPU周波数の設定を、ステップS5で決定された周波数に更新する(ステップS6)。

40

【0049】

そして、ステップS7へ進む。一方、仮想マシンのVMタイプ係数の値が0である場合(ステップS2:Yes)、またはCPU11の利用率がしきい値Min以上で、かつしきい値Max以下である場合(ステップS4:No)も、ステップS7へ進む。

【0050】

ゲスト電力制御モジュール31は、ゲスト電力制御管理表32にエントリのある全ての仮想マシンについて上述した一連の更新処理が終了したか否かを判断する(ステップS7

50

)。終了していれば(ステップS7: Yes)、一連のCPU周波数定期更新処理が終了となる。終了していなければ(ステップS7: No)、ゲスト電力制御モジュール31は、ゲスト電力制御管理表32を参照し、次の仮想マシンについてステップS2~ステップS7の処理を行う。ゲスト電力制御モジュール31は、以上の一連の処理を、ゲスト電力制御管理表32にエントリのある全ての仮想マシンについて終了するまで、繰り返す。

【0051】

[仮想マシン割り当て処理]

図6は、実施例2にかかる制御方法における仮想マシン割り当て処理を示すフローチャートである。仮想マシン割り当て処理は、CPUスケジューラ34により実施される。

【0052】

図6に示すように、仮想マシン割り当て処理は、例えば計算機に内蔵されたタイマーの割り込みにより起動される。CPU11にある仮想マシンが割り当てられて動作しているときに仮想マシン割り当て処理が開始されると、CPUスケジューラ34は、それまでCPU11で動作していた仮想マシンの状態をメインメモリ14やNANDメモリ15に保存する(ステップS11)。CPU11で動作していた仮想マシンの状態は、CPU11のレジスタから得られる。

【0053】

次いで、CPUスケジューラ34は、ポリシーに従って、次に動作させる仮想マシンを選択し、選択した仮想マシンの動作時間を決定する(ステップS12)。次いで、CPUスケジューラ34は、ステップS12で選択した仮想マシンの動作が開始された後の次のタイマーの割り込みタイミング、すなわちCPUスケジューラ34の起動タイミングを設定する(ステップS13)。

【0054】

次いで、CPUスケジューラ34は、CPU11のCPU周波数を切り替える処理を行う(ステップS14)。CPU11のCPU周波数を切り替える処理(CPU周波数切り替え処理)については後述する。次いで、CPUスケジューラ34は、ステップS12で選択した仮想マシンの状態を、以前に保存したメモリ領域から読み出し、CPU11のレジスタに仮想マシンの状態を設定して復元する(ステップS15)。これによって、CPUスケジューラ34によって新たにスケジューリングされた仮想マシンの実行が開始される。そして、CPUスケジューラ34は、仮想マシン割り当て処理を終了する。

【0055】

[仮想マシン切り替え時のCPU周波数切り替え処理]

図7は、実施例2にかかる制御方法におけるCPU周波数切り替え処理を示すフローチャートである。仮想マシン切り替え時のCPU周波数切り替え処理は、上述した仮想マシンの切り替え時にCPUスケジューラ34から呼び出される。

【0056】

図7に示すように、CPU周波数切り替え処理では、ゲスト電力制御モジュール31は、ゲスト電力制御管理表32から、切り替え前の仮想マシンのCPU周波数、及び切り替え後の仮想マシンのCPU周波数を参照する(ステップS21)。切り替え前の仮想マシンは、現在CPU11で動作中の仮想マシンである。切り替え後の仮想マシンは、次に動作する仮想マシンである。

【0057】

次いで、ゲスト電力制御モジュール31は、切り替え前の仮想マシンのCPU周波数と、切り替え後の仮想マシンのCPU周波数とが異なるか否かを判断する(ステップS22)。異なる場合(ステップS22: Yes)、ゲスト電力制御モジュール31は、CPU周波数変更処理モジュール35を呼び出す。呼び出されたCPU周波数変更処理モジュール35は、CPU11のCPU周波数の設定を、次に動作する仮想マシンのCPU周波数に更新する(ステップS23)。そして、CPUスケジューラ34による仮想マシン割り当て処理に戻る。

【0058】

10

20

30

40

50

一方、切り替え前の仮想マシンのCPU周波数と、切り替え後の仮想マシンのCPU周波数とが同じである場合も（ステップS22：No）、CPUスケジューラ34による仮想マシン割り当て処理に戻る。

【0059】

[フロント係数設定変更処理]

図8は、実施例2にかかる制御方法におけるフロント係数設定変更処理を示すフローチャートである。ユーザーによる画面の切り替え操作によって画面切り替え制御モジュール33が呼び出されると、画面切り替え制御モジュール33によってフロント係数設定変更処理が呼び出される。

【0060】

図8に示すように、フロント係数設定変更処理が開始されると、ゲスト電力制御モジュール31は、ゲスト電力制御管理表32を参照し、一番目の仮想マシンがフォアグラウンドで動作しているか否かを判断する（ステップS31）。フォアグラウンドで動作している場合（ステップS31：Yes）、ゲスト電力制御モジュール31は、ゲスト電力制御管理表32のフロント係数フィールドの、当該仮想マシンに対応する格納値をデフォルト値に設定する（ステップS32）。

10

【0061】

そして、ステップS34へ進む。一方、仮想マシンがフォアグラウンドで動作していない場合（ステップS31：No）、ゲスト電力制御管理表32のフロント係数フィールドの、当該仮想マシンに対応する格納値をデフォルト値よりも大きい値に設定する（ステップS33）。そして、ステップS34へ進む。

20

【0062】

ゲスト電力制御モジュール31は、ゲスト電力制御管理表32にエントリのある全ての仮想マシンについて上述した一連の変更処理が終了したか否かを判断する（ステップS34）。終了していれば（ステップS34：Yes）、一連のフロント係数設定変更処理が終了となる。終了していなければ（ステップS34：No）、ゲスト電力制御モジュール31は、ゲスト電力制御管理表32を参照し、次の仮想マシンについてステップS31～ステップS34の処理を行う。ゲスト電力制御モジュール31は、以上の一連の処理を、ゲスト電力制御管理表32にエントリのある全ての仮想マシンについて終了するまで、繰り返す。

30

【0063】

・動作の具体例

以下の各具体例では、図3に示す構成を例にして説明する。例えば、設定可能なCPU周波数を600MHz、500MHz、250MHz及び125MHzとして説明する。例えば、Maxデフォルト値を80%とし、Minデフォルト値を75%として説明する。また、CPU周波数を現在の設定値よりも高くする場合には、設定可能なCPU周波数の最も高い周波数が設定され、CPU周波数を現在の設定値よりも低くする場合には、設定可能なCPU周波数の一段低い周波数が設定されるとして説明する。なお、設定可能なCPU周波数、Maxデフォルト値及びMinデフォルト値は、これに限らないし、CPU周波数の設定値を変更する場合もこの例に限らない。

40

【0064】

[具体例1．CPUの利用率が下降していく例]

図9は、CPUの利用率が下降していく例における各種情報の一例を示す図表である。図9に示す図表において、各仮想マシンのしきい値Max及びしきい値Minは、上述した(1)式及び(2)式から算出したものである（図11、図13及び図15においても同じ）。具体例1では、ゲストVM-A21、ゲストVM-B22、ドライバVM-C23及びドライバVM-D24のCPU周波数が600MHzであり、サービスVM-E25のCPU周波数が250MHzである状態から説明する。

【0065】

図10は、CPUの利用率が下降していく例におけるCPU周波数の一例を示す模式図

50

である。直近のCPU利用率が100%である場合、ゲストVM - A 2 1、ゲストVM - B 2 2、ドライバVM - C 2 3及びドライバVM - D 2 4については、CPU利用率がしきい値Maxを超えているので、最も高い周波数である600MHzが設定される。サービスVM - E 2 5は、しきい値Max及びしきい値Minがともに0である。しきい値Max及びしきい値Minがともに0である仮想マシンのCPU周波数は、CPU利用率に関係なく、一定のままである。従って、サービスVM - E 2 5のCPU周波数は、250MHzのままである。

【0066】

直近のCPU利用率が85%になると、ドライバVM - D 2 4については、CPU利用率がしきい値Minよりも小さいので、一段低い周波数である500MHzが設定される。他の仮想マシンのCPU周波数は、変化しない。この場合、ゲスト電力制御管理表32のCPU周波数フィールドの、ドライバVM - D 2 4に対応する格納値が500MHzとなる。

10

【0067】

直近のCPU利用率が70%になると、ゲストVM - B 2 2、ドライバVM - C 2 3及びドライバVM - D 2 4については、CPU利用率がしきい値Minよりも小さいので、それぞれ、一段低い周波数である500MHz、500MHz及び250MHzが設定される。他の仮想マシンのCPU周波数は、変化しない。この場合、ゲスト電力制御管理表32のCPU周波数フィールドの、ゲストVM - B 2 2、ドライバVM - C 2 3及びドライバVM - D 2 4に対応する格納値は、それぞれ500MHz、500MHz及び250MHzとなる。

20

【0068】

直近のCPU利用率が65%になると、ゲストVM - A 2 1、ゲストVM - B 2 2、ドライバVM - C 2 3及びドライバVM - D 2 4については、CPU利用率がしきい値Minよりも小さいので、それぞれ、一段低い周波数である500MHz、250MHz、250MHz及び125MHzが設定される。サービスVM - E 2 5のCPU周波数は、変化しない。この場合、ゲスト電力制御管理表32のCPU周波数フィールドの、ゲストVM - A 2 1、ゲストVM - B 2 2、ドライバVM - C 2 3及びドライバVM - D 2 4に対応する格納値は、それぞれ500MHz、250MHz、250MHz及び125MHzとなる。

30

【0069】

このように、CPUの利用率が下降していく場合、バックグラウンドで動作しているゲストVM - BのCPU周波数が、フォアグラウンドで動作しているゲストVM - AのCPU周波数よりも優先的に低くなっていく。また、同じドライバVMでも、VMタイプ係数の値が大きい仮想マシンのCPU周波数が、優先的に低くなっていく。

【0070】

[具体例2 . CPUの利用率が上昇していく例]

図11は、CPUの利用率が上昇していく例における各種情報の一例を示す図表である。具体例2では、ゲストVM - A 2 1、ゲストVM - B 2 2、ドライバVM - C 2 3及びドライバVM - D 2 4のCPU周波数が125MHzであり、サービスVM - E 2 5のCPU周波数が250MHzである状態から説明する。

40

【0071】

図12は、CPUの利用率が上昇していく例におけるCPU周波数の一例を示す模式図である。直近のCPU利用率が65%である場合、ゲストVM - A 2 1、ゲストVM - B 2 2、ドライバVM - C 2 3及びドライバVM - D 2 4については、CPU利用率がしきい値Minよりも小さいので、一段低い周波数が設定される。ここでは、現在の設定値が最も低い125MHzであるので、125MHzが設定される。サービスVM - E 2 5のCPU周波数は、250MHzのままである。

【0072】

直近のCPU利用率が75%になると、ゲストVM - A 2 1については、CPU利用率

50

がしきい値 M_{max} を超えているので、最も高い周波数である 600MHz が設定される。他の仮想マシンの CPU 周波数は、変化しない。この場合、ゲスト電力制御管理表 3 2 の CPU 周波数フィールドの、ゲスト VM - A 2 1 に対応する格納値が 600MHz となる。

【0073】

直近の CPU 利用率が 85% になると、ゲスト VM - A 2 1、ゲスト VM - B 2 2 及びドライバ VM - C 2 3 については、CPU 利用率がしきい値 M_{max} を超えているので、最も高い周波数である 600MHz が設定される。他の仮想マシンの CPU 周波数は、変化しない。この場合、ゲスト電力制御管理表 3 2 の CPU 周波数フィールドの、ゲスト VM - A 2 1、ゲスト VM - B 2 2 及びドライバ VM - C 2 3 に対応する格納値が 600MHz となる。直近の CPU 利用率が 95% になった場合も同様である。

10

【0074】

このように、CPU の利用率が上昇していく場合、フォアグラウンドで動作しているゲスト VM - A 2 1 の CPU 周波数が、バックグラウンドで動作しているゲスト VM - B 2 2 の CPU 周波数よりも優先的に高くなっていく。また、同じドライバ VM でも、VM タイプ係数の値が小さい仮想マシンの CPU 周波数が、優先的に高くなっていく。

【0075】

[具体例 3 . フォアグラウンドで動作する仮想マシンを切り替える例]

図 1 3 は、フォアグラウンドで動作する仮想マシンを切り替える例における各種情報の一例を示す図表である。具体例 3 では、具体例 2 の図 1 1 と同様に、ゲスト VM - A 2 1、ゲスト VM - B 2 2、ドライバ VM - C 2 3 及びドライバ VM - D 2 4 の CPU 周波数が 125MHz であり、サービス VM - E 2 5 の CPU 周波数が 250MHz である状態から説明する。図 1 3 に示す各種情報は、フォアグラウンドで動作する仮想マシンを切り替えた後の情報である。

20

【0076】

図 1 4 は、フォアグラウンドで動作する仮想マシンを切り替える例における CPU 周波数の一例を示す模式図である。直近の CPU 利用率が 65% である場合、具体例 2 と同様に、ゲスト VM - A 2 1、ゲスト VM - B 2 2、ドライバ VM - C 2 3 及びドライバ VM - D 2 4 については、 125MHz の CPU 周波数が設定される。サービス VM - E 2 5 の CPU 周波数は、 250MHz のままである。

30

【0077】

直近の CPU 利用率が 75% になると、具体例 2 と同様に、ゲスト VM - A 2 1 の CPU 周波数は、 600MHz に設定される。他の仮想マシンの CPU 周波数は、変化しない。

【0078】

CPU 利用率が 75% のままであるが、フォアグラウンドで動作する仮想マシンがゲスト VM - A 2 1 からゲスト VM - B 2 2 に切り替わると、ゲスト VM - A 2 1 はバックグラウンドでの動作に切り替わる。それによって、図 1 3 に示すように、ゲスト VM - A 2 1 のフロント係数が 1 に更新され、ゲスト VM - B 2 2 のフロント係数が 0.9 に更新される。そして、ゲスト VM - A 2 1 のしきい値 M_{max} 及びしきい値 M_{min} は、それぞれ 80% 及び 75% に更新される。また、ゲスト VM - B 2 2 のしきい値 M_{max} 及びしきい値 M_{min} は、それぞれ 72% 及び 68% に更新される。

40

【0079】

従って、ゲスト VM - B 2 2 については、CPU 利用率がしきい値 M_{max} を超えることになるので、最も高い周波数である 600MHz が設定される。他の仮想マシンの CPU 周波数は、変化しない。この場合、図 1 3 に示すように、ゲスト電力制御管理表 3 2 の CPU 周波数フィールドの、ゲスト VM - B 2 2 に対応する格納値が 600MHz となる。

【0080】

直近の CPU 利用率が 70% になると、ゲスト VM - A 2 1、ドライバ VM - C 2 3 及びドライバ VM - D 2 4 については、CPU 利用率がしきい値 M_{min} よりも小さくなるの

50

で、一段低い周波数が設定される。ゲストVM - A 2 1のCPU周波数は、500MHzに設定される。ドライバVM - C 2 3及びドライバVM - D 2 4については、現在の設定値が最も低い125MHzであるので、125MHzが設定される(図14には現れていない)。他の仮想マシンのCPU周波数は、変化しない。この場合、ゲスト電力制御管理表32のCPU周波数フィールドの、ゲストVM - A 2 1に対応する格納値が500MHzとなる。

【0081】

このように、CPU利用率が上昇していくと、フォアグラウンドで動作しているゲストVM - A 2 1のCPU周波数が、バックグラウンドで動作しているゲストVM - B 2 2のCPU周波数よりも優先的に高くなっていく。そして、フォアグラウンドで動作する仮想マシンが切り替わると、新たにフォアグラウンドで動作することになったゲストVM - B 2 2のCPU周波数が、新たにバックグラウンドで動作することになったゲストVM - A 2 1のCPU周波数よりも優先的に高くなっていく。また、CPU利用率が下降してきたとき、バックグラウンドで動作するゲストVM - A 2 1のCPU周波数が、フォアグラウンドで動作しているゲストVM - B 2 2のCPU周波数よりも優先的に低くなっていく。

【0082】

[具体例4 . 新たに仮想マシンが起動する例]

図15は、新たに仮想マシンが起動する例における各種情報の一例を示す図表である。具体例4では、具体例1の図9と同様に、ゲストVM - A 2 1、ゲストVM - B 2 2、ドライバVM - C 2 3及びドライバVM - D 2 4のCPU周波数が600MHzであり、サービスVM - E 2 5のCPU周波数が250MHzである状態から説明する。図15に示す各種情報は、新たにVM_IDがFである仮想マシン(ゲストVM - Fとする)が起動した後の情報である。

【0083】

図16は、新たに仮想マシンが起動する例におけるCPU周波数の一例を示す模式図である。直近のCPU利用率が100%である場合、具体例1と同様に、ゲストVM - A 2 1、ゲストVM - B 2 2、ドライバVM - C 2 3及びドライバVM - D 2 4については、600MHzのCPU周波数が設定される。サービスVM - E 2 5のCPU周波数は、250MHzのままである。

【0084】

直近のCPU利用率が85%になると、具体例1と同様に、ドライバVM - D 2 4のCPU周波数が500MHzに設定される。他の仮想マシンのCPU周波数は、変化しない。

【0085】

ここで、CPU利用率が85%のままであるが、新たにゲストVM - Fが起動されたとする。ゲストVM - Fがバックグラウンドで動作すると、図15に示すように、CPU利用率がゲストVM - Fのしきい値Max(80%)を超えているので、ゲストVM - FのCPU周波数は、最も高い周波数である600MHzに設定される。

【0086】

直近のCPU利用率が70%になると、ゲストVM - B 2 2、ドライバVM - C 2 3、ドライバVM - D 2 4及びゲストVM - Fについては、CPU利用率がしきい値Minよりも小さいので、それぞれ、一段低い周波数である500MHz、500MHz、250MHz及び500MHzが設定される。他の仮想マシンのCPU周波数は、変化しない。この場合、ゲスト電力制御管理表32のCPU周波数フィールドの、ゲストVM - B 2 2、ドライバVM - C 2 3、ドライバVM - D 2 4及びゲストVM - Fに対応する格納値は、それぞれ500MHz、500MHz、250MHz及び500MHzとなる。

【0087】

直近のCPU利用率が65%になると、ゲストVM - A 2 1、ゲストVM - B 2 2、ドライバVM - C 2 3、ドライバVM - D 2 4及びゲストVM - Fについては、CPU利用率がしきい値Minよりも小さくなるので、一段低い周波数が設定される。ゲストVM -

10

20

30

40

50

A 2 1 の C P U 周波数は、5 0 0 M H z に設定される。ゲスト V M - B 2 2、ドライバ V M - C 2 3 及びゲスト V M - F の C P U 周波数は、2 5 0 M H z に設定される。ドライバ V M - D 2 4 の C P U 周波数は、1 2 5 M H z に設定される。サービス V M - E 2 5 の C P U 周波数は、2 5 0 M H z のままである。この場合、ゲスト電力制御管理表 3 2 の C P U 周波数フィールドの、ゲスト V M - A 2 1、ゲスト V M - B 2 2、ドライバ V M - C 2 3、ドライバ V M - D 2 4 及びゲスト V M - F に対応する格納値は、それぞれ 5 0 0 M H z、2 5 0 M H z、2 5 0 M H z、1 2 5 M H z 及び 2 5 0 M H z となる。

【 0 0 8 8 】

このように、新たに仮想マシンが起動しても、既に起動している仮想マシンと同様に C P U 周波数が制御される。

【 0 0 8 9 】

実施例 2 によれば、実施例 1 と同様の効果が得られる。

【 0 0 9 0 】

上述した実施例 1、2 に関し、さらに以下の付記を開示する。

【 0 0 9 1 】

(付記 1) 1 個以上の物理的なプロセッサと、前記物理的なプロセッサで実行されることによって実現される複数の仮想マシンごとに、前記仮想マシンの種別に応じて設定される、クロック周波数を制御する制御情報を管理する管理表と、前記仮想マシンごとに、前記管理表の制御情報に基づいてしきい値を求め、前記しきい値と前記物理的なプロセッサの利用率とに基づいてクロック周波数を決定する決定部と、前記決定部により決定されたクロック周波数に基づいて前記物理的なプロセッサのクロック周波数を変更する変更部と、を備えることを特徴とする計算機。

【 0 0 9 2 】

(付記 2) 前記制御情報は、前記仮想マシンごとに、前記仮想マシンがフォアグラウンドで動作しているか、バックグラウンドで動作しているか、に応じて設定される情報を含むことを特徴とする付記 1 に記載の計算機。

【 0 0 9 3 】

(付記 3) 前記決定部は、前記仮想マシンごとに、前記しきい値の最大値と最小値とを求め、前記物理的なプロセッサの利用率が前記しきい値の最大値よりも大きい場合にクロック周波数を上げ、前記物理的なプロセッサの利用率が前記しきい値の最小値よりも小さい場合にクロック周波数を下げるように決定することを特徴とする付記 1 または 2 に記載の計算機。

【 0 0 9 4 】

(付記 4) 前記決定部は、前記仮想マシンごとに、決定したクロック周波数に基づいて前記管理表を更新することを特徴とする付記 1 ~ 3 のいずれか一つに記載の計算機。

【 0 0 9 5 】

(付記 5) 前記仮想マシンの種別の一つは、前記物理的なプロセッサの利用率に連動してクロック周波数が変動する仮想マシンであることを特徴とする付記 1 ~ 4 のいずれか一つに記載の計算機。

【 0 0 9 6 】

(付記 6) 前記仮想マシンの種別の一つは、前記物理的なプロセッサの利用率に連動し、かつ重み付けされてクロック周波数が変動する仮想マシンであることを特徴とする付記 1 ~ 5 のいずれか一つに記載の計算機。

【 0 0 9 7 】

(付記 7) 前記仮想マシンの種別の一つは、一定のクロック周波数で動作する仮想マシンであることを特徴とする付記 1 ~ 6 のいずれか一つに記載の計算機。

【 0 0 9 8 】

(付記 8) 物理的なプロセッサの利用率を求め、前記物理的なプロセッサで実行されることによって実現される複数の仮想マシンごとに、前記仮想マシンの種別に応じて設定される、クロック周波数を制御する制御情報に基づいてしきい値を求め、前記仮想マシンごと

10

20

30

40

50

に、前記しきい値と前記物理的なプロセッサの利用率とに基づいてクロック周波数を決定することを特徴とする制御方法。

【0099】

(付記9) 前記決定されたクロック周波数に基づいて前記物理的なプロセッサのクロック周波数を変更することを特徴とする付記8に記載の制御方法。

【0100】

(付記10) 前記制御情報は、前記仮想マシンごとに、前記仮想マシンがフォアグラウンドで動作しているか、バックグラウンドで動作しているか、に応じて設定される情報を含むことを特徴とする付記8または9に記載の制御方法。

【0101】

(付記11) 前記仮想マシンごとに、前記しきい値の最大値と最小値とを求め、前記物理的なプロセッサの利用率が前記しきい値の最大値よりも大きい場合にクロック周波数を上げ、前記物理的なプロセッサの利用率が前記しきい値の最小値よりも小さい場合にクロック周波数を下げるように決定することを特徴とする付記8～10のいずれか一つに記載の制御方法。

【0102】

(付記12) 物理的なプロセッサの利用率を求め、前記物理的なプロセッサで実行されることによって実現される複数の仮想マシンごとに、前記仮想マシンの種別に応じて設定される、クロック周波数を制御する制御情報をメモリから読み出して前記制御情報に基づいてしきい値を求め、前記仮想マシンごとに、前記しきい値と前記物理的なプロセッサの利用率とに基づいてクロック周波数を決定する、処理をコンピュータに実行させるプログラム。

【0103】

(付記13) 前記決定されたクロック周波数に基づいて前記物理的なプロセッサのクロック周波数を変更することを特徴とする付記12に記載のプログラム。

【0104】

(付記14) 前記制御情報は、前記仮想マシンごとに、前記仮想マシンがフォアグラウンドで動作しているか、バックグラウンドで動作しているか、に応じて設定される情報を含むことを特徴とする付記12または13に記載のプログラム。

【0105】

(付記15) 前記仮想マシンごとに、前記しきい値の最大値と最小値とを求め、前記物理的なプロセッサの利用率が前記しきい値の最大値よりも大きい場合にクロック周波数を上げ、前記物理的なプロセッサの利用率が前記しきい値の最小値よりも小さい場合にクロック周波数を下げるように決定することを特徴とする付記12～14のいずれか一つに記載のプログラム。

【符号の説明】

【0106】

- 1, 2, 21～25 仮想マシン
- 3, 11 プロセッサ
- 4, 32 管理表
- 5, 31 決定部
- 6, 35 変更部

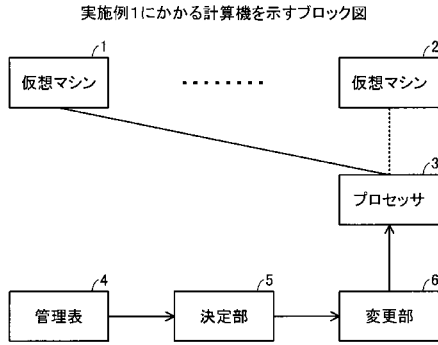
10

20

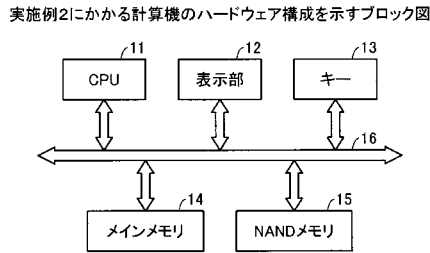
30

40

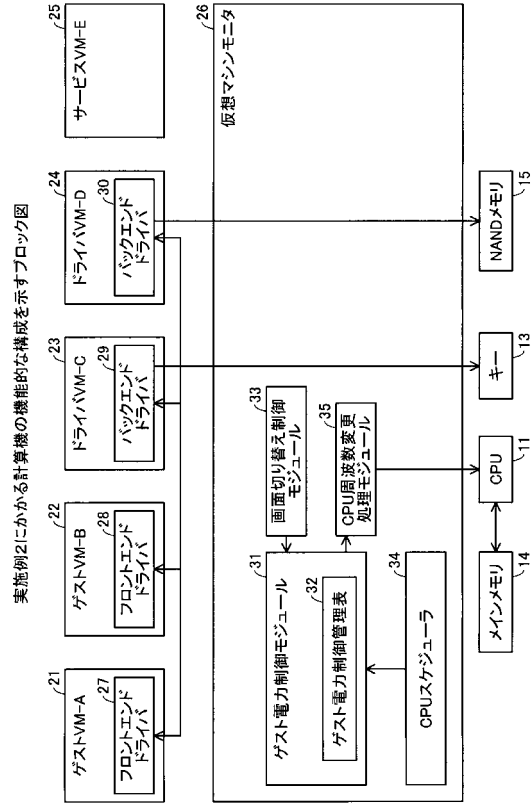
【 図 1 】



【 図 2 】



【 図 3 】



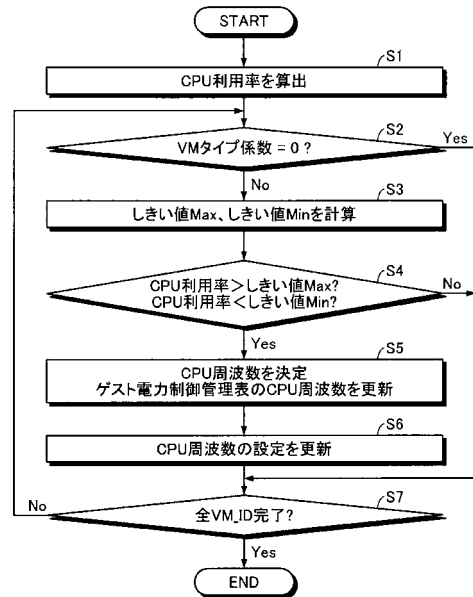
【 図 4 】

実施例2にかかる計算機の管理表の一例を示す図表

VM_ID	VMタイプ係数	フロント係数	CPU周波数
A (ゲストVM)	1	0.9	600 (MHz)
B (ゲストVM)	1	1	600 (MHz)
C (キードライバVM)	1	1	600 (MHz)
D (メモリドライバVM)	1.2	1	600 (MHz)
E (サービスVM)	0	1	250 (MHz)

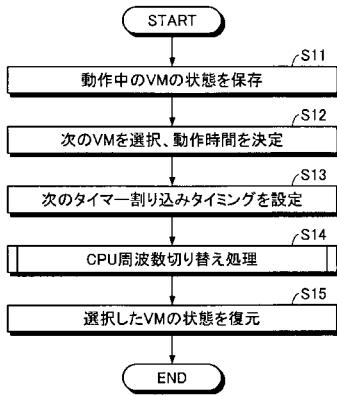
【 図 5 】

実施例2にかかる制御方法におけるCPU周波数定期更新処理を示すフローチャート



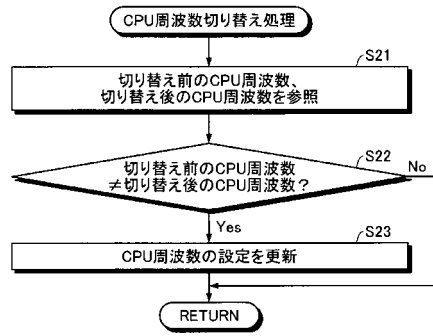
【 図 6 】

実施例2にかかる制御方法における
仮想マシン割り当て処理を示すフローチャート



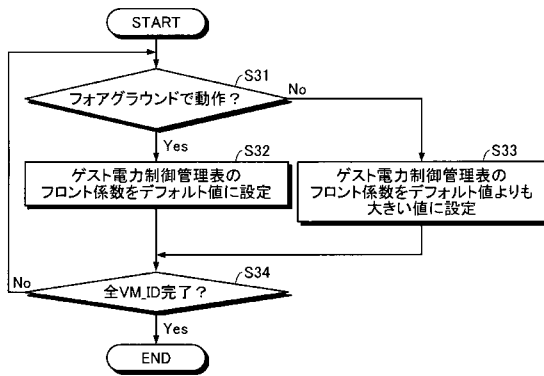
【 図 7 】

実施例2にかかる制御方法における
CPU周波数切り替え処理を示すフローチャート



【 図 8 】

実施例2にかかる制御方法における
フロント係数設定変更処理を示すフローチャート



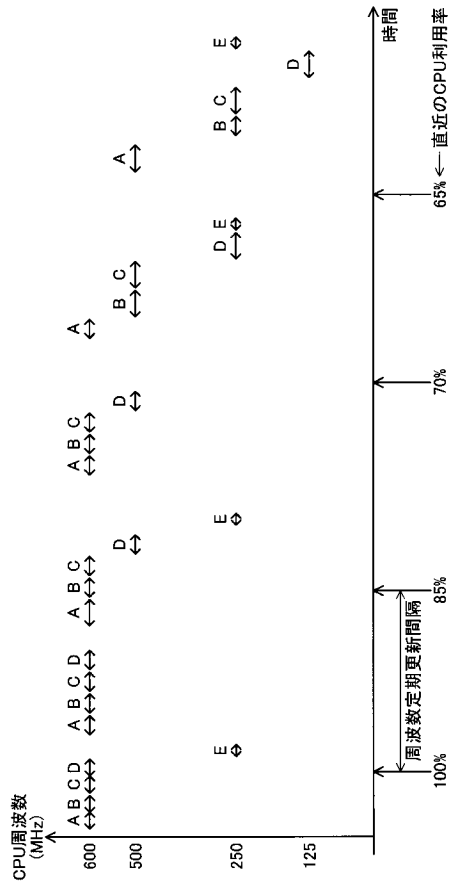
【 図 9 】

CPUの利用率が下降していく例における各種情報の一例を示す図表

VM_ID	VMタイプ係数	フロント係数	しきい値Max	しきい値Min	CPU周波数
A (ゲストVM)	1	0.9	72%	68%	600 (MHz)
B (ゲストVM)	1	1	80%	75%	600 (MHz)
C (キートドライバVM)	1	1	80%	75%	600 (MHz)
D (メモリドライバVM)	1.2	1	96%	90%	600 (MHz)
E (サーバーベースVM)	0	1	0	0	250 (MHz)

【 図 1 0 】

CPUの利用率が下降していく例におけるCPU周波数の一例を示す模式図



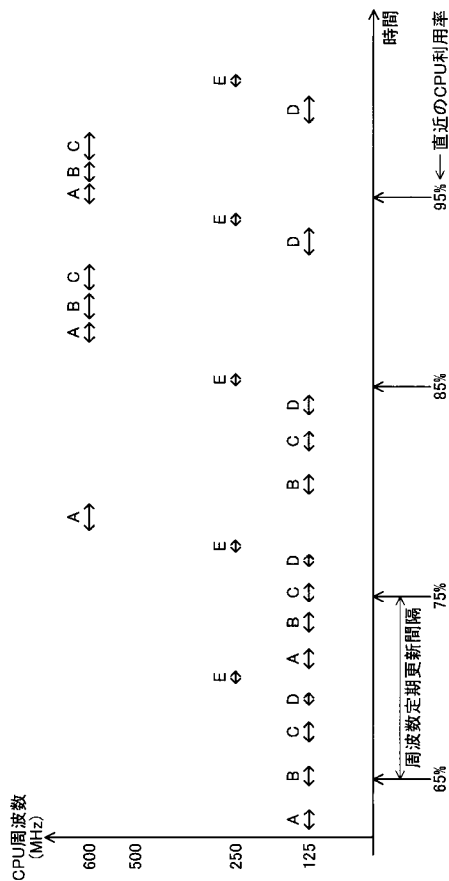
【 図 1 1 】

CPUの利用率が上昇していく例における各種情報の一例を示す図表

VM_ID	VMタイプ係数	フロント係数	しきい値Max	しきい値Min	CPU周波数
A (ゲストVM)	1	0.9	72%	68%	125 (MHz)
B (ゲストVM)	1	1	80%	75%	125 (MHz)
C (キードライバVM)	1	1	80%	75%	125 (MHz)
D (メモリアイバVM)	1.2	1	96%	90%	125 (MHz)
E (サーバースVM)	0	1	0	0	250 (MHz)

【 図 1 2 】

CPUの利用率が上昇していく例におけるCPU周波数の一例を示す模式図



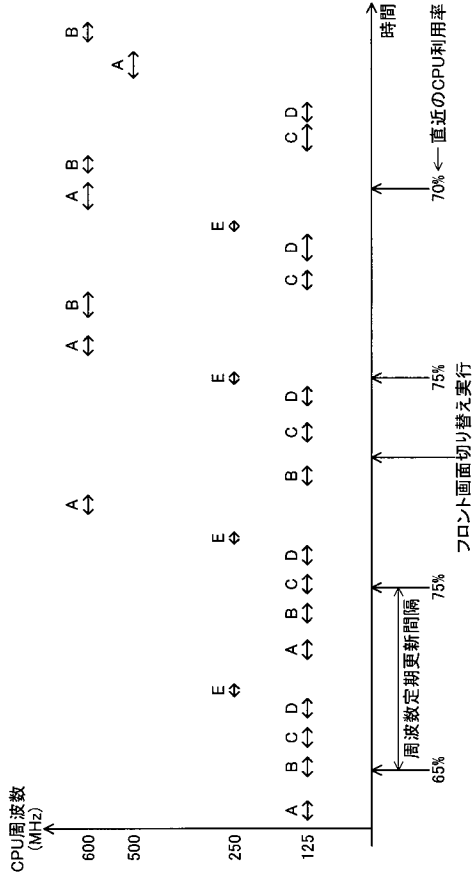
【 図 1 3 】

フォアグラウンドで動作する仮想マシンを切り替える例における各種情報の一例を示す図表

VM_ID	VMタイプ係数	フロント係数	しきい値Max	しきい値Min	CPU周波数
A (ゲストVM)	1	1	80%	75%	600 (MHz)
B (ゲストVM)	1	0.9	72%	68%	600 (MHz)
C (キードライバVM)	1	1	80%	75%	125 (MHz)
D (メモリアイバVM)	1.2	1	96%	90%	125 (MHz)
E (サーバースVM)	0	1	0	0	250 (MHz)

【 図 1 4 】

フォアグラウンドで動作する仮想マシンを切り替える例におけるCPU周波数の一例を示す模式図



【 図 1 5 】

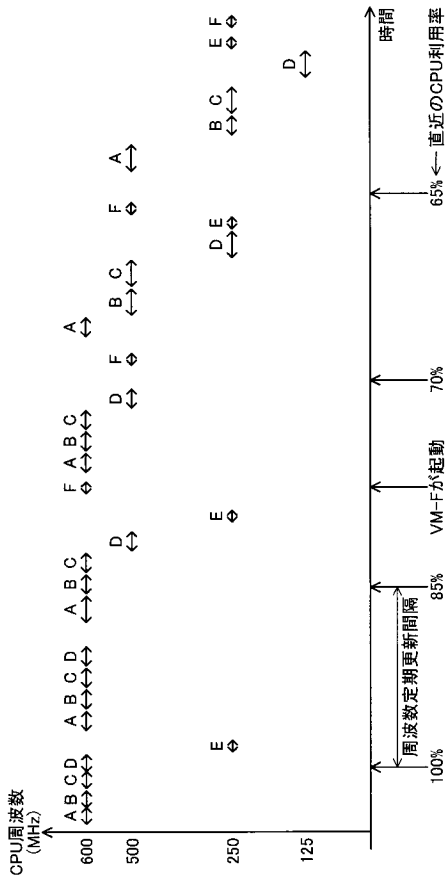
新たに仮想マシンが起動する例における各種情報の一例を示す図表

32

VM_ID	VMタイプ係数	フロント係数	しきい値Max	しきい値Min	CPU周波数
A (ゲストVM)	1	0.9	72%	68%	600 (MHz)
B (ゲストVM)	1	1	80%	75%	600 (MHz)
C (キードライバVM)	1	1	80%	75%	600 (MHz)
D (メモリドライバVM)	1.2	1	96%	90%	500 (MHz)
E (サービスVM)	0	1	0	0	250 (MHz)
F (ゲストVM)	1	1	80%	75%	600 (MHz)

【 図 1 6 】

新たに仮想マシンが起動する例におけるCPU周波数の一例を示す模式図



フロントページの続き

Fターム(参考) 5B011 LL11 LL13
5B079 AA07 BA01 BB01 BC01