

(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2013-131076

(P2013-131076A)

(43) 公開日 平成25年7月4日(2013.7.4)

(51) Int.Cl.  
G06F 11/34 (2006.01)F I  
G06F 11/34テーマコード (参考)  
5B042

審査請求 未請求 請求項の数 12 O L (全 29 頁)

(21) 出願番号 特願2011-280551 (P2011-280551)  
(22) 出願日 平成23年12月21日 (2011.12.21)(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 100108187  
弁理士 横山 淳一  
(72) 発明者 加藤 美保  
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 鈴木 啓之  
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 北山 翼  
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

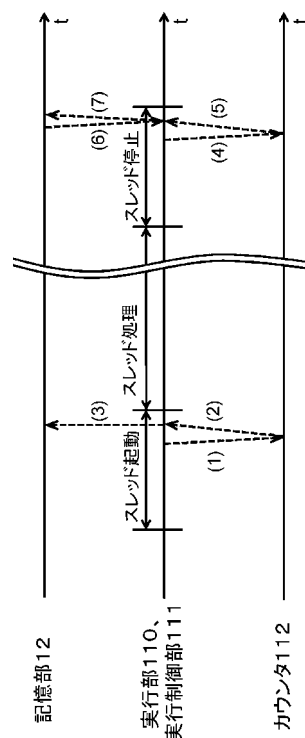
(54) 【発明の名称】 算出方法、算出プログラム及びコンピュータ

(57) 【要約】

【課題】一側面においては、供給されるクロック信号の周波数が変動するプロセッサへの割り当て制御対象である処理単位について、プロセッサの動作量を算出することを目的とする。

【解決手段】異なる周波数のクロック信号が供給され得るプロセッサを備えるコンピュータは、前記プロセッサへの割り当て制御の対象である処理単位の処理の開始制御に応じて、前記クロック信号を計数するクロックカウンタから読み出した第1の値と、前記処理の停止制御に応じて前記クロックカウンタから読み出した第2の値との差分を算出する、処理を実行する。

【選択図】 図4



**【特許請求の範囲】****【請求項 1】**

異なる周波数のクロック信号が供給され得るプロセッサを備えるコンピュータに、  
前記プロセッサへの割り当て制御の対象である処理単位の処理の開始制御に応じて、前記クロック信号のパルスを計数するクロックカウンタから読み出した第 1 の値と、前記処理の停止制御に応じて前記クロックカウンタから読み出した第 2 の値との差分を算出する、  
処理を実行させることを特徴とする算出プログラム。

**【請求項 2】**

前記コンピュータに、さらに、  
算出された前記差分に、単位クロック数当たりの所定種類のコスト値を示す係数を乗じて得られるコスト値を算出する、  
処理を実行させることを特徴とする請求項 1 に記載の算出プログラム。

10

**【請求項 3】**

前記処理の停止制御の実行は、前記処理の処理過程における前記プロセッサの処理状況に応じて判断される、  
ことを特徴とする請求項 1 または請求項 2 に記載の算出プログラム。

**【請求項 4】**

前記コンピュータに、さらに、  
算出した前記差分を、前記処理単位を識別する識別情報と関連付けて記憶部に記憶する、  
処理を実行させることを特徴とする請求項 1 ～請求項 3 のいずれか 1 つに記載の算出プログラム。

20

**【請求項 5】**

前記コンピュータに、さらに、  
前記処理単位の処理がさらに実行される場合に、さらに実行される処理の開始制御に応じて前記クロックカウンタから読みだされた第 3 の値と、前記さらに実行される処理の停止制御に応じて前記クロックカウンタから読みだされた第 4 の値との差分を算出し、  
前記第 3 の値と前記第 4 の値とに基づいて算出された前記差分を、前記識別情報と関連付けられて前記記憶部に記憶された前記差分に積算して得られる積算値を、前記識別情報に関連付けて前記記憶部に記憶する、  
処理を実行させることを特徴とする請求項 4 に記載の算出プログラム。

30

**【請求項 6】**

前記コンピュータに、さらに、  
他の処理単位が実行される場合に、前記他の処理単位の処理の開始制御に応じて前記クロックカウンタから読み出した第 5 の値と、前記他の処理単位の前記処理の停止制御に応じて前記クロックカウンタから読み出した第 6 の値との差分を算出し、  
前記他の処理単位が前記処理単位と同じプログラムについての処理単位である場合に、前記第 5 の値及び前記第 6 の値に基づいて算出された差分と、前記積算値との和を、前記プログラムを識別するプログラム識別情報に関連付けて前記記憶部に記憶する、  
処理を実行させることを特徴とする請求項 5 に記載の算出プログラム。

40

**【請求項 7】**

前記コンピュータに、さらに、  
前記プログラムに関する入力制御または出力制御のうち、少なくとも一方の制御を実行させるドライバプログラムの処理単位の処理が実行される場合に、前記ドライバプログラムの処理単位の処理の開始制御に応じて前記クロックカウンタから読み出した第 7 の値と、前記ドライバプログラムの前記処理単位の前記の停止処理の停止制御に応じて前記クロックカウンタから読み出した第 8 の値との差分を算出し、  
前記第 7 の値と前記第 8 の値とに基づく前記差分と、前記和との和を、前記プログラム識別情報と関連付けて前記記憶部に記憶する、

50

処理を実行させることを特徴とする請求項 6 に記載の算出プログラム。

【請求項 8】

前記処理単位は、マルチスレッド制御の対象となるスレッドであるか、マルチプロセス制御の対象となるプロセスであるか、または前記コンピュータへのアクセス元に関して行なわれる処理の単位である、

ことを特徴とする請求項 1 ～ 5 のいずれか 1 つに記載の算出プログラム。

【請求項 9】

異なる周波数のクロック信号が供給され得るプロセッサと、前記クロック信号を計数するクロックカウンタと、を含み、

前記プロセッサが、

前記プロセッサへの割り当て制御の対象である処理単位の処理の開始制御に応じて、前記クロックカウンタから読み出した第 1 の値と、前記処理の停止制御に応じて前記クロックカウンタから読み出した第 2 の値との差分を算出する、

処理を実行することを特徴とするコンピュータ。

【請求項 10】

前記クロックカウンタが、

前記クロック信号が前記プロセッサに供給されたパルス の数を計数する、

ことを特徴とする請求項 9 に記載のコンピュータ。

【請求項 11】

前記コンピュータは、さらにメモリを備え、

前記プロセッサが、

前記算出した前記差分を、前記処理単位を識別する識別情報と関連付けて前記メモリに記憶する、

処理を実行することを特徴とする請求項 9 または請求項 10 に記載のコンピュータ。

【請求項 12】

異なる周波数のクロック信号が供給され得るプロセッサ、及びクロックカウンタを含むコンピュータが、

前記クロックカウンタにより、前記クロック信号のパルスを計数し、

前記プロセッサへの割り当て制御の対象である処理単位の処理の開始制御に応じて、前記クロックカウンタから読み出した第 1 の値と、前記処理の停止制御に応じて前記クロックカウンタから読み出した第 2 の値との差分を算出する、

ことを実行させることを特徴とする算出方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、コンピュータに対して用いられる計測技術に関する。

【背景技術】

【0002】

1 つの物理マシンにおいて、複数の仮想マシンによる処理が実行されることがある。物理マシンで処理を実行する仮想マシンについての消費電力の推定を、仮想マシンの CPU (Central Processing Unit) 使用率に基づいて行なう技術がある (例えば、特許文献 1) 。

【先行技術文献】

【特許文献】

【0003】

【特許文献 1】特開 2010-039513 号公報

【発明の概要】

【発明が解決しようとする課題】

【0004】

CPU の使用率は、算出対象のプログラムについての処理により、どの程度の時間 CP

10

20

30

40

50

Uが占有されるかに基づいて算出される。一方で、CPUの消費電力や発熱を抑制させるため、CPUの処理負荷などに応じて、回路の同期に用いられる動作周波数（クロック周波数）を変動させる技術がある。CPUの動作周波数を変動させる制御が行なわれると、変動の前後においてCPUが実際に単位時間当たり動作した量が変化してしまう。CPUが動作することで電力が消費されるので、CPUの動作周波数を変動してCPUの動作量が変わると、CPUの消費電力も変動する。CPUを占有する時間に応じて消費電力を見積もる従来の方法では、動作周波数の変動による消費電力の変化が考慮されない。そのため、算出対象のプログラムについて、動作周波数の変動前後でCPUの占有時間が同じ場合には、CPUが実際に動作した量が変動しているにも関わらず、変動前後の消費電力が同じであると思われてしまう、という問題がある。

10

**【0005】**

本発明の一側面においては、異なる周波数のクロック信号が供給され得るプロセッサへの割り当て制御対象である処理単位について、プロセッサの動作量を算出することを目的とする。

**【課題を解決するための手段】****【0006】**

開示の算出プログラムは、異なる周波数のクロック信号が供給され得るプロセッサを備えるコンピュータに、前記プロセッサへの割り当て制御の対象である処理単位の処理の開始制御に応じて、前記クロック信号のパルスを計数するクロックカウンタから読み出した第1の値と、前記処理の停止制御に応じて前記クロックカウンタから読み出した第2の値との差分を算出する、処理を実行させる。

20

**【0007】**

開示のコンピュータは、異なる周波数のクロック信号が供給され得るプロセッサと、前記クロック信号を計数するクロックカウンタと、を含み、前記プロセッサが、前記プロセッサへの割り当て制御の対象である処理単位の処理の開始制御に応じて、前記クロックカウンタから読み出した第1の値と、前記処理の停止制御に応じて前記クロックカウンタから読み出した第2の値との差分を算出する、処理を実行する。

**【0008】**

開示の算出方法は、異なる周波数のクロック信号が供給され得るプロセッサ、及びクロックカウンタを含むコンピュータが、前記クロックカウンタにより、前記クロック信号のパルスを計数し、前記プロセッサへの割り当て制御の対象である処理単位の処理の開始制御に応じて、前記クロックカウンタから読み出した第1の値と、前記処理の停止制御に応じて前記クロックカウンタから読み出した第2の値との差分を算出する。

30

**【発明の効果】****【0009】**

本発明の一側面において、異なる周波数のクロック信号が供給され得るプロセッサへの割り当て制御対象である処理単位について、プロセッサの動作量を算出することができる。

**【図面の簡単な説明】****【0010】**

40

【図1】図1は、コンピュータ1の機能構成例を示す。

【図2】図2は、コンピュータ1のソフトウェア構成例を示す。

【図3】図3は、コンピュータ1のハードウェア構成例を示す。

【図4】図4は、スレッド制御に応じたデータ読出しの関係を示す。

【図5】図5は、スレッドコンテキストのデータ構造例を示す。

【図6】図6は、スレッド制御に応じたデータ読出しの関係を示す。

【図7】図7は、実行制御部111の制御フローを示す。

【図8】図8は、スレッド停止・終了の処理フローを示す。

【図9】図9は、スレッド起動・生成の処理フローを示す。

【図10】図10(a)－(c)は、プロセスとスレッドとの関係を示す。

50

- 【図 1 1】図 1 1 は、プログラムとプロセスとの関係を示す。
- 【図 1 2】図 1 2 は、プログラムごとの動作量を示す。
- 【図 1 3】図 1 3 は、プログラムごとの動作量算出のフローを示す。
- 【図 1 4】図 1 4 は、プログラムごとの動作量算出のフローを示す。
- 【図 1 5】図 1 5 は、計測結果を表示する画面情報の例を示す。
- 【図 1 6】図 1 6 は、コンピュータ 1 のソフトウェア構成例を示す。
- 【図 1 7】図 1 7 は、仮想マシンごとの動作量算出のフローを示す。
- 【図 1 8】図 1 8 は、仮想マシンとプログラムとの関係を示す。
- 【図 1 9】図 1 9 は、仮想マシンごとの動作量を示す。
- 【図 2 0】図 2 0 は、計測結果を表示する画面情報の例を示す。
- 【図 2 1】図 2 1 は、コンピュータ 1 のソフトウェア構成を示す。
- 【図 2 2】図 2 2 は、仮想マシンとプログラムとの関係を示す。
- 【図 2 3】図 2 3 は、仮想マシンごとの動作量を示す。
- 【図 2 4】図 2 4 は、システムの構成例を示す。
- 【図 2 5】図 2 5 は、コンピュータ 1 のソフトウェア構成例を示す。
- 【図 2 6】図 2 6 は、処理対象の単位の切り替えについてのシーケンスを示す。
- 【図 2 7】図 2 7 (a) - (c) は、プロセスとスレッドとの関係を示す。
- 【図 2 8】図 2 8 は、プログラムとプロセスとの関係を示す。
- 【図 2 9】図 2 9 は、プログラムごとの動作量を示す。
- 【発明を実施するための形態】

10

20

#### 【0011】

プロセッサの消費電力は、プロセッサに供給されるクロック信号の周波数の変動に応じて変動する。プロセッサは、1 周期分のクロック信号に応じてプロセッサ内の回路の充放電を 1 回行なうため、クロック信号の周波数の変動により単位時間当たりの充放電回数が増減するためである。クロック信号の周波数の変動は、例えば、クロック信号の供給元を、異なる周波数のクロック信号を生成する複数の発振器間で切り替えた場合や、1 つの発振器により生成されるクロック信号の周波数に経時的にズレが生じた場合などに起こる。

#### 【0012】

例えば、ある時間間隔  $t$  において、周波数  $f_1$  で動作した場合と、周波数  $f_2$  で動作した場合とにおけるプロセッサの充放電回数は、それぞれ  $f_1 \cdot t$ 、 $f_2 \cdot t$  となる。プロセッサが動作した時間で消費電力を見積もる方法があるが、時間でプロセッサの動作量を評価する場合には、数式 1 で示す  $t$  を越えた時間計測すると、充放電 1 回分以上の誤差が生じることとなる。言い換えると、ある時間範囲  $t$  におけるプロセッサの動作量を計測する場合には、クロックの周波数が数式 1 に示す  $(f_1 - f_2)$  以上に変動すると、クロック周期 1 回分以上の誤差が生じる。そのため、計測したい時間範囲  $t$  に対して周波数が数式 1 に示す  $(f_1 - f_2)$  以上に変動する場合においては、クロック信号の周波数の変動を考慮する必要があるが生じる。

30

$$t \leq \lceil 1 / (f_1 - f_2) \rceil \cdots \text{(数式 1)}$$

#### 【0013】

以下に本発明の実施形態について図面に基づいて説明する。まず、第 1 の実施形態のコンピュータ 1 の構成について、図 1 ~ 3 に基づいて説明する。

40

#### 【0014】

図 1 は、コンピュータ 1 の機能構成例を示す。コンピュータ 1 は、処理部 11 及び記憶部 12 を含み、処理部 11 は実行部 110、実行制御部 111、クロックカウンタ 112、読出処理部 113、算出部 114 及び記憶制御部 115 を含む。

#### 【0015】

図 2 は、コンピュータ 1 のソフトウェア構成例を示す。図 2 に示すソフトウェア構成例においては、アプリケーションプログラム 23a、アプリケーションプログラム 23b、アプリケーションプログラム 23c の複数種類のアプリケーションプログラムがコンピュータ 1 において動作する。コンピュータ 1 で動作するアプリケーションプログラムはアプ

50

リケーションプログラム 23 a-c に限定するものでなく、例えば、アプリケーションプログラム 23 a-c 以外のアプリケーションプログラムもコンピュータ 1 で動作しうる。さらに、コンピュータ 1 において、アプリケーション 23 a-c の処理をハードウェア (HW) 21 に実行させるための制御・管理を行なうオペレーションシステム (OS) 22 a が動作する。ハードウェア 21 は、図 3 を用いて後述されるハードウェアである。

#### 【0016】

図 3 は、コンピュータ 1 のハードウェア構成例を示す。コンピュータ 1 は、プロセッサ 301、RAM (Random Access Memory) 302、ROM (Read Only Memory) 303、ドライブ装置 304、記録媒体 305、通信インターフェース (I/F) 306、入力インターフェース (I/F) 307、出力インターフェース (I/F) 308 を含む。プロセッサ 301 は、プロセッサ 301 の動作に関連する回路との同期に用いられる動作周波数を変動させて動作する。プロセッサ 301、RAM 302、ROM 303、ドライブ装置 304、通信インターフェース 306、入力インターフェース 307、出力インターフェース 308 は、バス 309 と接続されている。通信インターフェース 306 は、ネットワーク 310 と接続されている。入力インターフェース 307 は、入力装置 311 と接続されており、出力インターフェース 308 は、出力装置 312 と接続されている。

#### 【0017】

図 1 ~ 3 に示されるコンピュータ 1 の各構成について説明する。

#### 【0018】

処理部 11 の機能は、例えば、プロセッサ 301 が OS 22 a、アプリケーションプログラム 23 a-c などに基づいて処理を行なうことで実現される。また、記憶部 12 の機能は、例えば、RAM 302 または記録媒体 305 などにプロセッサ 301 の制御により情報が読み書きされることで実現される。

#### 【0019】

実行部 110 は、後述される実行制御部 111 の制御に応じて、アプリケーションプログラム 23 a-c の処理を実行する。実行部 110 の機能は、例えば、記録媒体 305 から RAM 302 に読みだされたアプリケーションプログラム 23 a-c をプロセッサ 301 が読み込み、読み込んだプログラムに含まれる命令をプロセッサ 301 が処理することで実現される。

#### 【0020】

実行制御部 111 は、実行部 110 により行なわれるアプリケーションプログラム 23 a-c の処理のスケジューリングを行なう。各アプリケーションプログラムの処理をマルチプロセスで行なう場合には、実行制御部 111 は、各アプリケーションプログラムの処理単位であるプロセスをプロセッサ 301 に割り当てる制御を行なう。プロセスを割り当てるタイミングは、実行制御部 111 に含まれるプロセススケジューラのアルゴリズムに基づく処理状況についての判断により決められる。また、各アプリケーションプログラムの処理をマルチスレッドで行なう場合にも同様に、実行制御部 111 は、各アプリケーションプログラムの処理単位であるスレッドをプロセッサ 301 に割り当てる制御を行なう。スレッドを割り当てるタイミングは、実行制御部 111 に含まれるスレッドスケジューラのアルゴリズムに基づく処理状況についての判断により決められる。実行制御部 111 の機能は、例えば、プロセススケジューラ又はスレッドスケジューラを含む OS 22 a に基づいて、プロセッサ 301 と RAM 302 が動作することにより実現される。

#### 【0021】

クロックカウンタ 112 は、プロセッサ 301 の処理において回路の同期に用いられるクロックが動作する周期に応じて加算されるカウンタであり、プロセッサ 301 に備えられている。プロセッサ 301 がマルチコアであれば、各コアについてカウンタが備えられる。マルチコアである場合には、各コアにおける動作量は、各コアに設けられたカウンタに基づいて計測される。以下の説明において特にことわらない限り、プロセッサ 301 が 1 つのコアを含む場合を例に挙げて説明する。クロックカウンタ 112 は、あるタイミン

10

20

30

40

50

グにおいて、プロセッサ 301 がそのタイミングまでにどれだけの動作を行なったかを示す。例えば、クロックカウンタ 112 は、ハードウェア 21 の起動時に計数を開始し、プロセッサ 301 の動作周波数で時間が 1 周期経過するたびに（1 パルスごとに）、クロックカウンタ 112 用に設けられたレジスタに例えば 1 bit ずつ加算するなどの制御を行なう。クロックカウンタ 112 の値を格納するレジスタは、例えば、32 bit や 64 bit などのサイズの領域を設ける。以下の説明においては、読出し命令により読みだされたクロック値が、変数  $clk1$  として扱われる。

#### 【0022】

読出処理部 113 は、実行制御部 111 によるスケジューリングのうち、プロセス又はスレッドを生成、起動（実行状態にさせる）、停止（休眠状態にさせる）又は終了させる制御に応じて、プロセッサ 301 に備えられたクロックカウンタ 112 から、クロック値を読み出す処理を行なう。読出し処理部 113 の機能は、例えば、プロセッサ 301 が、スケジューリング制御を実行させる命令群に読出し命令を追加された OS 22a に基づく制御を実行することにより実現される。

10

#### 【0023】

この読出し命令は、プロセッサ 301 に備えられたクロックカウンタ 112 の値を読み出す命令であって、例えば、Performance monitoring 機構（CPU 種毎に実装）などである。

#### 【0024】

算出部 114 は、実行制御部 111 の制御に応じて、読出処理部 113 により読みだされたクロック値（ $clk1$ ）を用いた演算（数式 2～4 参照）を行なう。例えば、算出部 114 は、実行制御部 111 によるプロセス又はスレッドの生成又は起動に応じて読出処理部 113 がクロック値（ $clk1$ ）を読み出した場合には、変数  $clk0$  にクロック値（ $clk1$ ）を代入する演算（数式 2 参照）を行なう。また、算出部 114 は、実行制御部 111 によるプロセス又はスレッドの停止又は終了に応じて読出処理部 113 がクロック値（ $clk1$ ）を読み出した場合には、クロック値（ $clk1$ ）と、値（ $clk0$ ）との差分  $d$  を算出する（数式 3 参照）。さらに、例えば、算出部 114 は、クロック値（ $clk1$ ）と値（ $clk0$ ）との差分  $d$  を積算値  $S$  に積算する演算（数式 4 参照）を行なう。この積算値  $S$  は、後述の記憶制御部 115 により、記憶部 12 に対して書込みと読出しが行なわれる値である。値（ $clk0$ ）と積算値  $S$  の初期値は、例えば 0 などの値である。また、数式 2～4 に含まれる「=」は、等号ではなく、代入を示す演算子である。

20

30

$$clk0 = clk1 \quad \dots \quad (\text{数式 2})$$

$$d = clk1 - clk0 \quad \dots \quad (\text{数式 3})$$

$$S = S + d \quad \dots \quad (\text{数式 4})$$

#### 【0025】

算出部 114 の機能は、OS 22a によるプロセス又はスレッドのスレッド制御（スレッドの生成、起動、停止、終了の制御）の際に、プロセッサ 301 にクロック値（ $clk1$ ）に基づいて、数式 2～4 の演算を実行させることにより実現される。クロック値  $clk0$ 、クロック値  $clk1$  は、それぞれプロセッサ 301 内のレジスタなどに格納される。

40

#### 【0026】

記憶制御部 115 は、実行制御部 111 に応じて記憶部 12 に記憶されたデータの読み出し、書き込みを行なう。例えば、記憶制御部 115 は、記憶部 12 から、プロセス又はスレッドの停止又は終了制御に応じて、停止又は終了制御の対象のプロセス又はスレッドのコンテキストを読み出す。コンテキストは、それぞれのプロセス又はスレッドに関する情報であり、それぞれのプロセス又はスレッドに対応する積算値  $S$  を含む。この記憶制御部 115 の機能は、プロセッサ 301 が、RAM 302 に格納された、停止又は終了制御の対象となるプロセス又はスレッドに対応するプロセスコンテキスト又はスレッドコンテキストから積算値  $S$  を読み出すことで実現される。

#### 【0027】

50

また、例えば、記憶制御部 115 は、算出部 114 によって算出された積算値 S と、実行制御部 111 で停止又は終了されるプロセス又はスレッドとを関連付けた情報と、を記憶部 12 に記憶させる。記憶制御部 115 の機能は、プロセッサ 301 が、プロセッサ 301 の数式 3 及び数式 4 の演算により算出された積算値 S を、RAM 302 に格納された、停止又は終了されるプロセス又はスレッドに対応するプロセスコンテキスト又はスレッドコンテキストに書き込むことにより実現される。さらには、例えば、実行制御部 111 によるプロセスまたはスレッドの起動または生成においては、記憶制御部 115 は、プロセッサ 301 の数式 2 の演算により算出された値  $clock$  を、起動または生成されるプロセスコンテキストまたはスレッドコンテキストに格納する。

#### 【0028】

10

また、他の方法では、記憶制御部 115 は、算出部 114 が差分  $d$  を算出するたびに、算出された差分  $d$  を対応するプロセスコンテキストまたはスレッドコンテキストに格納する。同じプロセスまたはスレッドが複数回実行された場合には、差分  $d$  が同じプロセスコンテキストまたはスレッドコンテキストに複数格納される。

#### 【0029】

処理部 11 に含まれる実行部 110、実行制御部 111、クロックカウンタ 112、読出処理部 113、算出部 114 及び記憶制御部 115 の上記の処理により、プロセッサ 301 で実行されるプロセス又はスレッドごとに、プロセス又はスレッドの処理に要したクロック数が計測される。

#### 【0030】

20

図 3 に示した各ハードウェアについて、さらに以下に説明する。

#### 【0031】

RAM 302 は読み書き可能なメモリ装置であって、例えば、SRAM (Static RAM) や DRAM (Dynamic RAM) などの半導体メモリ、または RAM でなくてもフラッシュメモリなどが用いられる。ROM 303 は、PROM (Programmable ROM) などを含む。ドライブ装置 304 は、記録媒体 305 に記録された情報の読み出しや書き込みかの少なくともいずれか一方を行なう装置である。記録媒体 305 は、ドライブ装置 304 によって書き込まれた情報を記憶する。記録媒体 305 は、例えば、ハードディスク、CD (Compact Disc)、DVD (Digital Versatile Disc)、ブルーレイディスクなどの記録媒体である。また、例えば、コンピュータ 1 は、複数種類の記録媒体それぞれについて、ドライブ装置 304 及び記録媒体 305 を設ける。

30

#### 【0032】

入力装置 311 は、操作に応じて入力信号を送信する装置である。入力信号は、例えば、キーボードやコンピュータ 1 の本体に取り付けられたボタンなどのキー装置や、マウスやタッチパネルなどのポインティングデバイスである。出力装置は、コンピュータ 1 の制御に応じて情報を出力する装置である。出力装置は、例えば、ディスプレイなどの画像出力装置や、スピーカーなどの音声出力装置などである。また、タッチスクリーンなどの入出力装置は、例えば、入力装置 311 及び出力装置 312 として用いられる。

#### 【0033】

40

図 4～6 に基づいて、スレッド制御の際のデータ読出しの関係について説明する。図 4～6 においては、マルチスレッド制御が行なわれる場合について説明しているが、マルチプロセス制御が行なわれる場合であっても、データ読出しの関係は同様である。

#### 【0034】

図 4 は、実行部 110 及びクロックカウンタ 112 と、記憶部 12 に格納されたスレッドコンテキストとのデータ読出しの関係を示す。

#### 【0035】

実行制御部 111 は、スレッドスケジューラに基づいて、スレッドの起動を行なう (スレッドが存在しない場合にはスレッドの生成を行なう)。実行制御部 111 のスレッド起動処理中に読出処理部 113 は、クロックカウンタ 112 からクロック数の読み出し処理

50



を行なう（矢印（１）、（２））。読み出したクロック数（ $clk_1$ ）は、算出部１１４及び記憶制御部１１５の処理により、起動時クロック数（ $clk_0$ ）として、図５に示すスレッドコンテキストＴ１に書き込まれる（ $clk_0$ の書き込み：矢印（３））。スレッドの起動が完了すると、実行部１１０は、図５に示すスレッドコンテキストに含まれる情報に基づいて、起動されたスレッドについての処理を行なう。

#### 【００３６】

図５は、スレッドコンテキストＴ１のデータ構造の一例を示す。スレッドコンテキストＴ１は、プロセッサ３０１がスレッドの処理に用いられる処理情報と、スレッドの管理情報と、を含む。処理情報は、プログラムカウンタ、スタックポインタなどのデータ読み出し位置を指定する情報である。実行部１１０は、スレッドコンテキストＴ１中の情報に指定されたプログラムカウンタ、スタックポインタなどに基づいて処理を行なう。スレッドの管理情報は、スレッドのＩＤ、時刻情報（スレッドが起動された時刻 $t_0$ 、スレッド処理時間の積算値 $S_t$ など）、クロック数情報（スレッドが起動された際のクロック数 $clk_0$ 、スレッド処理に要したクロック数の積算値 $S$ ）を含む。

#### 【００３７】

マルチプロセス制御に用いるプロセスコンテキストは、例えば、図５に示すスレッドコンテキストと同様のデータ構造図を有し、スレッドＩＤではなくプロセスＩＤを含む。また、プロセスコンテキストは、例えば、さらに、プロセス内で動作するスレッドを示す情報を含む。

#### 【００３８】

実行制御部１１１は、他のスレッドへの切り換えなどにより、スレッドの停止処理を行なう。記憶制御部１１５は、スレッドの停止処理において、プログラムカウンタ、スタックポインタなどの処理情報をスレッドコンテキストＴ１に保存する。スレッド停止処理中に、読出処理部１１３は、クロックカウンタ１１２からクロック数の読み出し処理を行なう（矢印（４）、（５））。算出部１１４及び記憶制御部１１５は、読み出したクロック数（ $clk_1$ ）と、起動処理時にスレッドコンテキストＴ１に書き込んだクロック数（ $clk_0$ の取得：矢印（６））とに基づいて、積算値 $S$ を算出して、スレッドコンテキストＴ１に書き込む（矢印（７））。

#### 【００３９】

コンピュータ１は、上述の処理により、矢印（１）（２）の読み出し時点から、矢印（４）（５）の読み出し時点までのクロック数をスレッドコンテキストに反映させることができる。

#### 【００４０】

図６は、スレッドが複数存在する場合のデータ読出しの関係を示す。図６の「スレッド $a$ 」で示す期間は、スレッド $a$ についてのスレッド起動、スレッドの処理及びスレッド停止の処理に要する期間を示す。

#### 【００４１】

スレッド $a$ の起動及び停止における記憶制御部１１５の書き込み処理は、スレッドコンテキスト（ $a$ ）に対して、矢印（１）（２）で読みだしたクロック数に基づいて行なわれる。スレッドコンテキスト（ $a$ ）は、スレッド $a$ に関するスレッドコンテキストである。また、スレッド $b$ の起動及び停止における記憶制御部１１５の書き込み処理は、スレッドコンテキスト（ $b$ ）に対して、矢印（３）（４）で読みだしたクロック数に基づいて行なわれる。スレッドコンテキスト（ $b$ ）はスレッド $b$ に関するスレッドコンテキストである。さらに、スレッド $b$ についての処理後に、再度スレッド $a$ の処理が行なわれた場合には、スレッド $a$ の起動及び停止における記憶制御部１１５の書き込み処理は、スレッドコンテキスト（ $a$ ）に対して、矢印（５）（６）で読みだしたクロック数に基づいて行なわれる。上記の記憶制御部１１５によりスレッドコンテキストに含まれる積算値 $S$ が更新されることで、スレッドごとに処理に要したクロック数が計測される。

#### 【００４２】

また、本実施形態において、プロセッサ３０１の動作周波数は変動する。例えば、矢印

10

20

30

40

50

Xで示した時点でプロセッサの動作周波数が変化したとすると、矢印（１）（２）間と矢印（５）（６）間とは、プロセッサ３０１の動作周波数が異なる。すなわち、単位時間当たりの動作量が異なる。したがって従来のプロセッサ３０１を占有する時間を計測する方法により、矢印（１）（２）間の時間と矢印（５）（６）間の時間とに基づいてプロセッサ３０１の負荷を評価すると、単位時間あたりの実際の動作量が異なるので不整合が生じてしまう。しかしながら、本実施形態においては、動作周波数が変動する場合にも、変動前後の矢印（１）（２）間と矢印（５）（６）間とについて、実際にプロセッサ３０１が動作した量を計測できる。

#### 【００４３】

図７は、スレッドごとに処理に要したクロック数をカウントするための処理部１１の制御フローを示す。この制御は、例えば、プロセッサ３０１がＯＳ２２ａに基づいて行なう制御である。プロセスごとに処理に要するクロック数をカウントする場合も図７と同様のフローが行なわれる。ただし、その場合はスレッドではなく、プロセスを対象として制御が行なわれる。

#### 【００４４】

まず実行制御部１１１は、スレッド制御のイベントが発生したか否かを判断する（Ｓ１１）。実行制御部１１１によりイベントが発生したと判断された場合（Ｓ１１：Ｙｅｓ）に、実行制御部１１１は、スレッドの終了または停止の処理（Ｓ１２）を開始する。このＳ１２の処理については後述する。また、Ｓ１２の処理が終わると、実行制御部１１１は、スレッド生成または起動の処理（Ｓ１３）を開始する。Ｓ１３の処理についても後述する。スレッド制御のイベントを発生させる判断は、実行中のスレッドの処理が完了する場合や、他のスレッドによる割り込みが発生した場合などのプロセッサ３０１に割り当てられる処理の状況に応じて、ＯＳ２２ａに含まれるマルチスレッドのスケジューラによって行なわれる。

#### 【００４５】

図８は、前述のスレッド停止または終了の処理（Ｓ１２）の処理例を示す。まず、実行制御部１１１は、発生したイベントがスレッド停止であるか否かを判断する（Ｓ２１）。発生したイベントが、スレッド停止ではない場合（Ｓ２１：Ｎｏ）には、実行制御部１１１は、発生したイベントがスレッド終了であるか否かを判断する（Ｓ２９）。イベントの発生やイベントの発生に応じて行なう処理は、ＯＳ２２ａに含まれるマルチスレッドのスケジューラによって、割り込みが起きたか否か、スレッドの処理が終了したか、などの状況に基づいて判断される。

#### 【００４６】

発生したイベントがスレッド停止である場合（Ｓ２１：Ｙｅｓ）には、実行制御部１１１は、プロセッサ３０１が処理中のスレッドについて命令の読み出しを行なうことを停止する（Ｓ２２）。Ｓ２２の処理後、実行制御部１１１は、命令の読み出しを停止されたスレッドに関する処理情報を取得する。例えば、実行制御部１１１は、プロセッサ３０１のプログラムカウンタとスタックポインタの値を読み出す。さらに、実行制御部１１１は、記憶制御部１１５に、読み出した処理情報を、記憶部１２に記憶された、停止処理中のスレッドに対応するスレッドコンテキストに記憶させる制御を行なう（Ｓ２３）。

#### 【００４７】

Ｓ２３の処理が行なわれた場合に、読出処理部１１３、算出部１１４及び記憶制御部１１５の処理により、Ｓ２４～Ｓ２７の停止処理中のスレッドのクロック数の更新処理が行なわれる。まず、読出処理部１１３が、クロックカウンタ１１２からクロック値（ｃｌｋ１）を読み出し（Ｓ２４）、記憶制御部１１５が、停止処理中のスレッドに対応するスレッドコンテキストから、起動時クロック値（ｃｌｋ０）と積算値Ｓとを読み出す（Ｓ２５）。このＳ２５の処理は、Ｓ２４よりも先に行なわれてもよい。

#### 【００４８】

算出部１１４は、Ｓ２４、Ｓ２５で読み出したクロック値（ｃｌｋ１）と起動時クロック値（ｃｌｋ０）と積算値Ｓとを用いて、数式３及び数式４の演算を行なって、積算値Ｓ

10

20

30

40

50

を更新する（S 2 6）。記憶制御部 1 1 5 は、更新された積算値 S を、記憶部 1 2 の停止処理中のスレッドに対応するスレッドコンテキストに記憶させる（S 2 7）。S 2 7 の処理を実行すると、S 1 2 の処理が終了する（S 2 8）。

【0 0 4 9】

S 2 9 の判断の結果、発生したイベントがスレッド終了である場合（S 2 9 : Y e s）には、読出処理部 1 1 3、算出部 1 1 4 及び記憶制御部 1 1 5 の処理により、S 3 0 ~ S 3 3 の終了処理中のスレッドのクロック数の更新処理が行なわれる。S 3 0 ~ S 3 3 の処理は、S 2 4 ~ S 2 7 と同様の処理である。

【0 0 5 0】

S 3 3 の処理後、記憶制御部 1 1 5 は、終了処理中のスレッドに対応するスレッドコンテキストに含まれるスレッド I D と積算値 S とを、記憶部 1 2 に含まれる別の記憶領域に記憶させる（S 3 4）。これにより、スレッドコンテキストの記憶領域が解放されても、終了されたスレッドについての動作量を示すデータを残しておくことができる。S 3 4 の処理が終了した場合に、S 1 2 の処理が終了する（S 2 8）。

【0 0 5 1】

S 2 9 の判断の結果、スレッド終了でない場合（S 2 9 : N o）には、S 1 2 の処理が終了する（S 2 8）。例えば、コンピュータ 1 が起動した直後にスレッドを立ち上げる場合などには、事前に処理中のスレッドがないため、S 2 9 の判断においてスレッド終了でないという判断が行なわれる。

【0 0 5 2】

図 9 は、前述のスレッド生成または起動の処理（S 1 3）の処理例を示す。まず、実行制御部 1 1 1 は、発生したイベントがスレッド起動であるか否かを判断する（S 4 1）。発生したイベントがスレッド停止ではない場合（S 4 1 : N o）には、実行制御部 1 1 1 は、発生したイベントがスレッド終了であるか否かを判断する（S 4 7）。イベントの発生やイベントの発生に応じて行なう処理は、O S 2 2 a に含まれるマルチスレッドのスケジューラによって、割り込みが起きたか否か、スレッドの処理が終了したか、などの状況に基づいて判断される。

【0 0 5 3】

発生したイベントがスレッド起動である場合（S 4 1 : Y e s）には、読出処理部 1 1 3 は、クロック値（c l k 1）をクロックカウンタ 1 1 2 から読み出す（S 4 2）。S 4 2 の処理の処理によりクロック値（c l k 1）が読み出されると、算出部 1 1 4 は数式 2 に基づいて、変数 c l k 0 にクロック値（c l k 1）を代入する。記憶制御部 1 1 5 は、値を代入された c l k 0 を、起動させるスレッドに対応するスレッドコンテキストに記憶させる（S 4 3）。

【0 0 5 4】

さらに、記憶制御部 1 1 5 は、記憶部 1 2 から、記憶させるスレッドに対応するスレッドコンテキスト内の処理情報を読み出す（S 4 4）。処理情報は、起動させるスレッドの処理に用いられる情報であり、例えばプログラムカウンタやスタックポインタの値などである。S 4 4 の処理は、S 4 2 や S 4 3 の処理と前後しても構わない。S 4 4 の処理が行なわれると、実行制御部 1 1 1 は、S 4 4 で読み出した処理情報に基づく処理を実行部 1 1 0 に開始させる（S 4 5）。S 4 5 の処理を行なうと、S 1 3 の処理は終了する（S 4 6）。

【0 0 5 5】

S 4 7 の判断の結果、発生したイベントがスレッド生成である場合（S 4 7 : Y e s）には、実行制御部 1 1 1 はスレッドコンテキストの生成を行なう（S 4 8）。S 4 8 の処理において、実行制御部 1 1 1 の指示に基づいて、記憶制御部 1 1 5 は、記憶部 1 2 にスレッドコンテキストを記憶する記憶領域を決定し、生成されるスレッドのスレッド I D をその記憶領域内に記憶させる。

【0 0 5 6】

また、読出処理部 1 1 3 は、クロックカウンタ 1 1 2 からクロック値（c l k 1）を読

10

20

30

40

50

み出す（S49）。S49の処理は、S48の処理よりも前に行われてもよい。S49の処理の処理によりクロック値（clk1）が読み出されると、算出部114は数式2に基づいて、変数clk0にクロック値（clk1）を代入する。記憶制御部115は、値を代入されたclk0を、起動させるスレッドに対応するスレッドコンテキストに記憶させる（S50）。S50の処理が行なわれると、実行制御部111は、実行部110による処理を開始させる（S51）。S51の処理を行なうと、S13の処理は終了する（S46）。S13の処理が終了すると、実行制御部111は、S11の処理を行ない、スレッド制御のイベントの発生を検知する。

#### 【0057】

コンピュータ1が上記の処理を実行することにより、コンピュータ1で動作するスレッドまたはプロセスごとに、プロセッサが処理に要したクロック数が計測される。プロセッサの動作により電力が消費されるので、プロセッサが実際に動作した量を計測することは、プロセッサの消費電力などのプロセッサの動作に要したコストの推定に有用である。例えば、占有時間に応じてプロセッサの消費電力を見積もる従来の方法では、動作周波数の変動前後で消費電力に不整合が生じてしまう。その一方、上記の処理によれば、プロセッサの実際の動作量を取得しているため、プロセッサの動作周波数に変動があっても変動した動作周波数に応じて消費される電力値を見積もることができる。本実施形態では、プロセスやスレッドごとに、処理に要したクロック数を測定することができるので、プロセスやスレッドごとの消費電力の推定に有用な動作量が把握可能となる。また、コンピュータ1は、プロセッサの動作に要したコストの量として、消費電力以外にも、例えば、プロセッサが消費する電力の発電により生じた二酸化炭素の量、プロセッサから発熱される熱量などについてプロセッサの動作量に応じて推定する。

#### 【0058】

消費電力の推定値は、例えば、プロセッサの1クロック当たりの消費電力量を予め算出しておき、本実施形態により計測したクロック数と、予め算出した1クロック当たりの消費電力量との乗算により算出される。二酸化炭素量の推定値についても同様に、例えば、予め算出したプロセッサの1クロック当たりの排出二酸化炭素量と、本実施形態により計測したクロック数との乗算により算出される。また、例えば、排出二酸化炭素量の算出において、プロセッサに共有される電力の発電形態に応じて、本実施形態により計測されるクロック数に乗じる値の切り替えが行なわれる。例えば、化石燃料を用いた発電と、自然エネルギーを用いた発電とで、2種類の値が用いられる。プロセッサから発生する熱量についても同様に、例えば、予め算出したプロセッサの1クロック当たりの発熱量を示す値をクロック数に乗算するなどの演算により求めることができる。

#### 【0059】

また、プロセッサは、プロセッサ内の回路への充放電に加えてリーク電流によっても電力を消費する。そのため、一つの例としては、実行部110は、電力消費を算出する場合に、リーク電流による消費電力量を、クロック数に基づいて求められた消費電力量に合算し、合算した値を出力する。リーク電流による消費電力量は、例えば、処理部10によるクロック数の計数と同様の処理により、プロセスやスレッドごとに、処理に要した時間が計測され、計測された時間に、実行部110が、リーク電流による単位時間当たりの消費電力量を乗算することで算出される。その際、処理部10に含まれる読出部113は、クロックカウンタ112からクロック数を読み出す際に、タイマから時刻情報を取得する。さらに、実行制御部111、記憶制御部115は、クロック数の計数と同様の処理を行ない、プロセスコンテキストまたはスレッドコンテキストに、処理に要した積算時間を格納する。

#### 【0060】

第2の実施形態について、以下に説明する。第2の実施形態においては、第1の実施形態のコンピュータ1が用いられる。第2の実施形態においては、コンピュータ1において動作するアプリケーションプログラム23cにより、アプリケーションプログラム23a、23bの処理によるプロセッサ301の動作量の可視化処理が行なわれる。実行部11

0 は、アプリケーションプログラム 23c に基づいて、例えば以下に説明する可視化処理を行なう。

#### 【0061】

図10(a)－(c)は、プロセスとスレッドの関係を示すテーブルT2a－T2cを示す。図10(a)のテーブルT2aは、プロセスIDが「A－1」であるプロセスが、スレッドID「a」のスレッドを含むプロセスであることを示す。図10(b)のテーブルT2bは、プロセスIDが「B－1」であるプロセスが、スレッドID「b」のスレッドと、スレッドID「c」のスレッドと、を含むプロセスであることを示す。図10(c)のテーブルT2cは、プロセスIDが「A－2」であるプロセスが、スレッドID「d」のスレッドを含むプロセスであることを示す。図10(b)に示すように、1つのプロセスにおいて、2つ以上のスレッドが生成されることもある。プロセスとスレッドの関係を示すテーブルは、図10(a)－(c)に限らず、コンピュータ1で動作するプロセスそれぞれについて生成される。

10

#### 【0062】

図11は、プログラムとプロセスとの関係を示すテーブルを示す。テーブルT2には、コンピュータ1で実行中または過去に実行されたプログラムと、各プログラムの処理単位であるプロセスとを対応付けた情報である。テーブルT2及びテーブルT3は、記憶部12に記憶される。図11では、プログラムID「A」とプロセスID「A－1」及び「A－2」とが関連付けられており、プログラムID「B」とプロセスID「B－1」及び「B－2」とが関連付けられており、プログラムID「C」とプロセスID「C－1」が関連付けられている。ちなみに、アプリケーションプログラム23aのプログラムIDが「A」であり、アプリケーションプログラム23bのプログラムIDが「B」であり、アプリケーションプログラム23cのプログラムIDが「C」であるとする。

20

#### 【0063】

実行部110は、コンピュータ1で動作するプログラム、プロセス、スレッドの情報を抽出し、プロセスとスレッドとの関係を示すテーブルT2と、プログラムとプロセスの関係を示すテーブルT3と、を更新する。この更新は、定期的、コンピュータ1に行なわれる操作に応じたタイミング、予め定められたタイミングなどのタイミングで行なわれる。実行部110は、更新するタイミングになると、各プロセスのプロセスコンテキスト内に格納されたスレッドIDを取得し、取得したスレッドIDを各プロセスに対応するテーブルT2に格納することで、テーブルT2を更新する。また、実行部110は、更新するタイミングになると、実行制御部111がプロセス制御に用いるプロセス管理テーブル（不図示）に格納されたプロセスIDを取得し、取得したプロセスIDを各プログラムに対応づけてテーブルT3に格納して、テーブルT3を更新する。

30

#### 【0064】

さらに、実行部110は、コンピュータ1で実行された各スレッドコンテキスト又は各プロセスコンテキスト内の積算値Sを収集して、収集した積算値SとテーブルT2、T3とに基づいてコンピュータ1で動作するプログラムそれぞれの動作量を算出する。プロセスコンテキストまたはスレッドコンテキストに、積算値Sでなく、差分dを複数格納している場合には、プロセスコンテキストまたはスレッドコンテキスト内に記憶された差分dの総和を算出し、その総和を積算値Sとして取り扱う。各プログラムについての動作量は、例えば図12に示すテーブルT4で管理される。テーブルT4においては、各プログラムのプログラムIDと、そのプログラムについてのプログラム動作量Tとが関連付けられている。

40

#### 【0065】

図13は、各プログラムのプログラム動作量Tを算出するフローを示す。この処理は、実行部110がアプリケーションプログラム23cに従って行なう処理である。プログラム動作量Tの算出は、テーブルT2、T3の更新と同様に、定期的、コンピュータ1に行なわれる操作に応じたタイミング、予め定められたタイミングなどのタイミングで行なわれる。S61の処理においては、そのタイミングであるか否かが判断される。実行部11

50

0 は、プログラム動作量 T 算出のタイミングであると判断された場合 (S 6 1 : Y e s) に、テーブル T 3 に示されるプログラム I D のうち、未選択のプログラム I D を選択する (S 6 2)。実行制御部 1 1 0 は、S 6 2 で選択されたプログラム I D についての動作量を示す変数 D をクリアする (S 6 3)。S 6 3 の処理は S 6 2 の処理よりも前に行なわれてもよい。

#### 【0066】

次に、実行部 1 1 0 は、テーブル T 3 で、S 6 2 で選択されたプログラム I D に関連づけられたプロセス I D のうち、未選択のプロセス I D を選択する (S 6 4)。S 6 4 の処理後、実行部 1 1 0 は、テーブル T 2 で、S 6 4 で選択されたプロセス I D に関連付けられたスレッド I D のうち、未選択のスレッド I D を選択する (S 6 5)。

10

#### 【0067】

実行部 1 1 0 は、S 6 5 で選択されたスレッド I D に対応するスレッドコンテキストから積算値 S を読み出す (S 6 6)。S 6 6 の処理後、実行部 1 1 0 は、変数 D の値を、S 6 6 で読み出した積算値 S と変数 D との合計に更新する (S 6 7)。すなわち、実行部 1 1 0 は、 $D = D + S$  の演算を行なう。この場合の「=」は代入を示す演算子である。この際、実行部 1 1 0 はスレッドコンテキストに含まれる積算値 S をクリア ( $S = 0$ ) する (S 6 8)。

#### 【0068】

S 6 8 の処理後、実行部 1 1 0 は、テーブル T 2 で、S 6 4 で選択されたプロセス I D に関連付けられたスレッド I D のうち、未選択のスレッド I D がないかどうかを判断する。S 6 8 の判断の結果、未選択のスレッド I D がある場合 (S 6 9 : N o) に、実行部 1 1 0 は、再度 S 6 5 の処理を行なう。S 6 9 の判断の結果、未選択のスレッド I D がない場合 (S 6 9 : Y e s) に、実行部 1 1 0 は、テーブル T 3 で、S 6 2 で選択されたプログラム I D に関連付けられたプロセス I D のうち、未選択のプロセス I D がないかどうかを判断する (S 7 0)。S 7 0 の判断の結果、未選択のプロセス I D がある場合 (S 7 0 : N o) に、実行部 1 1 0 は、再度 S 6 4 の処理を行なう。

20

#### 【0069】

S 7 0 の判断の結果、未選択のプロセス I D がない場合 (S 7 0 : Y e s) に、実行部 1 1 0 は、テーブル T 4 で、選択されたプログラム I D に関連付けられたプログラム動作量 T を更新する (S 7 1)。すなわち、実行部 1 1 0 は、 $T = T + D$  の演算を行なう。この場合の「=」は代入を示す演算子である。S 7 2 の処理後、実行部 1 1 0 は、テーブル T 3 に示されるプログラム I D のうち、未選択のプログラム I D がないかどうかを判断する。S 7 2 の判断の結果、未選択のスレッド I D がある場合 (S 7 2 : N o) に、実行部 1 1 0 は、再度 S 6 2 の処理を行なう。

30

#### 【0070】

S 7 2 の判断の結果、未選択のプログラムがない場合 (S 7 2 : Y e s) に、実行部 1 1 0 は、表示処理を要求されたか否かを判断する (S 7 3)。これは、例えば、ユーザから表示要求を入力された場合、他の装置から表示要求を受信した場合などに、表示処理が要求されたと判断する。表示処理が要求されない場合 (S 7 3 : N o) には、実行部 1 1 0 は、処理を終了する (S 7 5)。

40

#### 【0071】

S 7 3 の判断の結果、表示処理が要求されたと判断された場合 (S 7 3 : Y e s) に、例えば、図 1 5 に示す画面情報 4 1 を生成する (S 7 4)。実行部 1 1 0 は、ユーザから表示要求が受信された場合には、生成された画面情報 4 1 を出力装置 3 1 2 に表示させ、他の装置から表示要求を受信した場合には、表示要求の送信元の装置に画面情報 4 1 を送信する。図 1 5 に示す画面情報 4 1 は、各アプリケーションについてのクロック数と、消費電力と、二酸化炭素消費量と、を示す情報を含む。各アプリケーションについての消費電力と、二酸化炭素排出量と、については、第 1 の実施形態で説明したように、例えば、1 クロック当たりの消費電力値、二酸化炭素排出量のそれぞれについてプログラム動作量 T を乗じることで算出される。

50

## 【0072】

第3の実施形態について以下に説明する。第3の実施形態においては、第1の実施形態のコンピュータ1のハードウェアが用いられる。第3の実施形態は、第1の実施形態、第2の実施形態とソフトウェア構成が異なる。

## 【0073】

図16は、第3の実施形態におけるコンピュータ1のソフトウェア構成例を示す。コンピュータ1で複数の仮想マシン26a-cが動作し、各仮想マシンの処理を実行するためのハードウェア21の制御がハイパーバイザ22bにより行なわれる。

## 【0074】

第3の実施形態においても、コンピュータ1は、実行部110、実行制御部111、クロックカウンタ112、読出し処理部113、算出部114、記憶制御部115を含む。第3の実施形態のハイパーバイザ22bに基づく処理により、プロセッサ301は、第1の実施形態の実行制御部111、読出し処理部113、算出部114及び記憶制御部115の機能を実現する。第3の実施形態においては、各仮想マシンについての処理の処理単位であるプロセスやスレッドについてのプロセッサ301の動作量が、ハイパーバイザの制御によりカウントされる。

## 【0075】

第3の実施形態においては、仮想マシン26cは、動作量管理の仮想マシンである。その場合の仮想マシン26cに基づく実行部110の処理について、図17に基づいて以下に説明する。

## 【0076】

まず、実行部110は、第2の実施形態におけるS61～S72の処理と同様に、アプリケーションプログラムごとに動作量を計測する処理を行なう（S81）。次に、実行部110は、図18に示す仮想マシン管理テーブルT5に含まれる仮想マシンの識別情報（VMID）のうち、未選択のVMIDを選択する（S82）。ここで、仮想マシン26aのVMIDを「X」とし、仮想マシン26bのVMIDを「Y」とし、仮想マシン26cのVMIDを「Z」とする。S82の処理が行なわれると、仮想マシンごとの動作量を示す変数Uをクリアする（S83）。さらに、実行部110は、図18の仮想マシン管理テーブルT5において、S82で選択されたVMIDに関連付けられたプログラムのうち、未選択のプログラムを選択する（S84）。実行部110は、S84で選択されたプログラムについてのプログラム動作量TをテーブルT4から読み出し、読みだしたプログラム動作量Tに基づいて変数Uを更新する。例えば、実行部110は、変数Uとプログラム動作量Tとの和を、変数Uに代入して変数Uを更新する。

## 【0077】

S84の処理を行なうと、実行部110は、図18の仮想マシン管理テーブルT5において、S82で選択されたVMIDに関連付けられたプログラムのうち、未選択のプログラムが無いかどうかを判断する（S86）。S86の判断の結果、未選択のプログラムがある場合（S86：No）には、実行部110は、再度S84の処理を行なう。

## 【0078】

S86の判断の結果、未選択のプログラムがない場合（S86：Yes）には、実行部110は、変数Uの値と、S82で選択したVMIDと、を関連付けて図19に示すテーブルT6に記憶する（S87）。S87の処理後、実行部110はテーブルT5に含まれるVMIDのうち、未選択のVMIDがないかどうか判断する（S88）。S88の判断の結果、未選択のVMIDがある場合（S88：No）には、実行部110は、S82の処理を再度行なう。

## 【0079】

S88の処理の結果、未選択のVMがない場合（S88：Yes）には、実行部110は、第2の実施形態のS73、S74の処理を行なう（S89）。例えば、要求に応じて画面情報を生成し、生成した画面情報の送信、または生成した画面情報の表示処理を行なう。

## 【0080】

図20に、第3の実施形態における画面情報の例を示す。画面情報42は、仮想マシン26aを利用するユーザ向けに仮想マシン26の動作量を示す画面情報である。図20に示すごとく、各月の使用状況を含む画面情報を生成するために、例えば、実行部110は、各月の終わりにテーブルT6を別の記憶領域にコピーし、テーブルT6に含まれるVM動作量の情報をクリアすることを行なう。さらに、実行部110は、画面情報を生成する場合には、表示対象のVMIDについて、テーブルT6に記憶されたVM動作量と、別の記憶領域に記憶された過去の月のVM動作量と、を読み出す。第2の実施形態と同様に、実行部110は、読み出したVM動作量のそれぞれに、1クロック当たりの消費電力量、1クロック当たりの二酸化炭素排出量を乗算して、各月の消費電力と二酸化炭素排出量とを算出する。各月の課金額は、例えば、消費電力と同様に1クロック当たりの課金額とクロック数との乗算により算出された値、予め行なわれた契約などで定められる。

10

## 【0081】

第4の実施形態について以下に説明する。第4の実施形態においては、第3の実施形態と同様に仮想マシンがコンピュータ1で動作し、コンピュータ1で動作する各仮想マシンの処理を行なうハードウェア21を制御するドライバの機能を有する仮想マシンが、さらにコンピュータ1で動作する。一方、第3の実施形態においては、例えば、ハイパーバイザ22bがドライバ機能を有している。また、第3の実施形態と同様に、仮想マシン26cは、例えば、動作量管理の仮想マシンなどである。

## 【0082】

20

図21は、第4の実施形態のコンピュータ1におけるソフトウェア構成例を示す。図16に示す第3の実施形態のソフトウェア構成に加えて、ドライバVM27aとドライバVM27bとが動作している。VM26aに関する処理を行なうハードウェアはドライバVM27aに基づいて制御される。また、VM26bに関する入出力を行なうハードウェアはドライバVM27bに基づいて制御される。

## 【0083】

図22のテーブルT7は、第4の実施形態の仮想マシン管理テーブルを示す。仮想マシン管理テーブルT7は、コンピュータ1で動作する仮想マシン（ドライバVMを含む）のIDと、各仮想マシンで動作するプログラムのIDと、を関係づけたデータを格納するテーブルである。仮想マシン管理テーブルにおいて、ドライバVM27aのIDは「XX」であり、ドライバVM27bのIDは「YY」である。また、仮想マシン管理テーブルT7では、VM間対応関係も示している。例えば、仮想マシン管理テーブルT7は、対応関係を示す情報として、VMID「XX」と関連付けられたVMID「X」を含む。この情報により、VMIDが「XX」である仮想マシンが、VMID「X」の仮想マシンについての処理を行なう仮想マシンであることが示される。

30

## 【0084】

第4の実施形態においては、仮想マシン26cに基づいて実行部110が図17のフローを実行する。第3の実施形態と異なり、S82では、ドライバVM26a、26b、26cも選択される。実行部110がS82～S88のフローを実行することで、コンピュータ1で動作する各仮想マシンについての動作量が算出される。図23は、各仮想マシンについての動作量を示す。さらに、図17のS89のフローにおいて、実行部110は、前述の表示処理を行なう。第4の実施形態においては、実行部110は、仮想マシン管理テーブルT7に対応関係を示される仮想マシンについては、対応関係を示された仮想マシン同士の動作量の和を算出して、算出した和を表示する。

40

## 【0085】

第4の実施形態で、上記の処理を行なうことで、プロセッサにより行なわれる計測対象のプログラムに関する演算処理に加え、プロセッサにより行なわれるハードウェア制御についての動作量も計測できる。また、消費電力を算出する際に乗じる係数は、例えば、仮想マシン26a～cの動作量に乗じる係数とドライバVM27a、bに乗じる係数とで、異なる値を用いる。その場合には、例えば、さらに、ドライバVM27a、bについて各

50



ドライバに制御される各ハードウェアの消費電力も含めた係数が設定される。

#### 【0086】

以下に第5の実施形態について説明する。第5の実施形態は、例えば、第1の実施形態のコンピュータ1を含む図24のシステムにおいて実施される形態である。

#### 【0087】

図24に例示されるシステムは、コンピュータ1と、コンピュータ2と、中継装置3と、中継装置4と、ネットワーク5と、を含む。コンピュータ1は、第1～4の実施形態のコンピュータ1と同様のハードウェア構成と機能構成を有する。コンピュータ2は、例えば、コンピュータ1と同様のハードウェア構成を有するコンピュータである。中継装置3は、コンピュータ1とネットワーク5との間で通信の中継を行なう装置である。中継装置3は、コンピュータ2とネットワーク5との間で通信の中継を行なう装置である。また、例えば、コンピュータ2が、コンピュータ1で動作する仮想マシンによりサービスを提供されるコンピュータである。また、サービスを提供されるコンピュータは、図24に示されるコンピュータ2でなくても、例えば、不図示の複数のコンピュータに対してサービスが提供される。

10

#### 【0088】

第5の実施形態においては、サービスの提供先ごとにプロセッサ301の動作量の計測を行なう。サービスの提供先ごとの計測とは、サービスの提供を要求するアクセス元に関して行なわれる処理ごとに行なわれるプロセッサ301の動作量についての計測である。例えば、サービスの認証を受けたユーザ単位（ユーザIDなどで識別されるアカウント単位）や、サービスに関する要求の送信元IPアドレスなどの単位で行なわれる処理についてのコンピュータ1の動作量の計測である。

20

#### 【0089】

図25は、コンピュータ1のソフトウェア構成例を示す。コンピュータ1では、第3の実施形態と同様に、ハイパーバイザ22bと、仮想マシン26a～cが動作する。第5の実施形態においては、仮想マシン26aは、OS28aとアプリケーションプログラム29aとアプリケーションプログラム29bとを含み、仮想マシン26bは、OS28bとアプリケーションプログラム29cとアプリケーションプログラム29dとを含む。仮想マシン26a、26bともに、図25に示すソフトウェア構成に限定するものでなく、他のアプリケーションプログラムもコンピュータ1で動作しうる。

30

#### 【0090】

第5の実施形態においては、アプリケーションプログラム29aに基づいて、他のコンピュータへのサービスの提供を行なわれ、アプリケーションプログラム29bに基づいて仮想マシン26aに含まれるアプリケーションプログラムによる動作量の管理が行なわれる。また、OS28aは、第1の実施形態のOS22aと同様に、OS28a上で動作するアプリケーションプログラムに応じたハードウェア制御を行なうためのソフトウェアであって、OS28a上で動作する各アプリケーションプログラムについて動作量をカウントさせる。しかしながら、第5の実施形態では、第1の実施形態と異なり、OS28aによる制御は、ハードウェア21をエミュレートするハイパーバイザ22bに対する制御となる。ハードウェア21は、ハイパーバイザ22bの制御に応じて処理を実行する。

40

#### 【0091】

図26は、計測対象となる単位の切り替えタイミングを示すシーケンス図である。すなわち、図26に示される切り替タイミングとは、アプリケーションプログラム29aにおける処理の対象となる単位の切り替え、OS28aにおける処理対象のアプリケーションプログラムの切り替え、HV22bにおける処理対象のVMの切り替え、などである。アプリケーションプログラム29aにおける処理の対象となる単位とは、サービス（アプリケーションプログラム26aにより提供される）によって識別される単位（ユーザ（アカウント）、アドレスなどの処理の対象となる単位）で、図26においては、CLIDという識別情報で識別される。識別情報CLIDは、例えば、ユーザID、処理要求の送信元アドレスなどである。

50

## 【0092】

ハイパーバイザ22bにより、処理対象のVMの切り替えが行なわれる。ハイパーバイザ22bにより、VM26aの処理が行なわれる期間に、OS28aが処理対象とするアプリケーションプログラムの処理が行なわれる。

## 【0093】

OS28bにより、処理対象のアプリケーションの切り替えが行なわれる。処理対象のアプリケーションの切り替えとは、実質的に、第1の実施形態において説明したように、プロセスまたはスレッドの切り替えにより行なわれる。OS28bの制御により、プロセスまたはスレッドの切り替え（プロセスまたはスレッドについての停止、終了、起動、生成などを含む）に応じて、クロックカウンタの値が読み出される。このとき読み出されるクロックカウンタの値は、ハイパーバイザ22bによりエミュレートされたクロックカウンタであり、VM26aに関する処理に要したクロック数をカウントして得られる値である。

10

## 【0094】

アプリケーションプログラム29aにより、処理対象の単位の切り替えが行なわれる。図26においては、識別情報CLIDがC1の単位から、C2の単位に切り替えられる（X1）。アプリケーションプログラム29aは、切り替え（X1）において、OSに対する切り替えの通知を実行させる。切り替えの通知は、切り替え先の単位の識別情報CLIDを含む。アプリケーション28aから切り替えの通知に応じて、OS28aによって、クロックのカウント処理及びカウンタ値の記憶場所変更が行なわれる。クロックのカウント処理については、アプリケーションの切り替えに応じたカウントと同様に行なわれる（図7～9参照）。カウンタ値の記憶場所変更は、クロック数の積算値Sを格納するスレッドコンテキストまたはプロセスコンテキストを切り替える処理である。後述するが、第5の実施形態においては、スレッドコンテキスト及びプロセスコンテキストは、処理対象の単位ごとに設けられる。

20

## 【0095】

図27、28は、第5の実施形態におけるプログラム、プロセス及びスレッドを管理するための管理テーブル例を示す。第5の実施形態においては、例えば、プログラム、プロセス及びスレッドが、処理対象ごとに管理される。すなわち、図5、11及び12に示すテーブルに含まれるプログラム、プロセス及びスレッドのIDのそれぞれに処理対象の識別情報CLIDが付加されて管理される。

30

## 【0096】

図27（a）～（c）は、プロセスとスレッドとの対応関係を示す。第5の実施形態においては、OS28aにより、プロセスIDとスレッドIDとの双方に、通知された識別情報CLIDが付加される。図28は、プログラムとプロセスとの対応関係を示す。第5の実施形態においては、OS28aにより、プログラムIDとプロセスIDとの双方に、通知された識別情報CLIDが付加される。

## 【0097】

OS28aにより、記憶場所変更が行なわれるにあたって、切り替え先の処理対象の単位について、図27、図28に示される情報がない場合については、アプリケーションプログラム29aからの通知に含まれる識別情報CLIDに基づいて、管理のための情報を生成する。例えば、図27、図28に示すプログラムとプロセスとの関係及びプロセスとスレッドとの関係を示す情報と、CLIDそれぞれに対応するスレッドコンテキスト及びプロセスコンテキストと、が生成される。

40

## 【0098】

また、アプリケーションプログラム29bは、第2の実施形態のアプリケーションプログラム23cと同様に、例えば、OS28a上で動作するアプリケーションの動作量管理のアプリケーションプログラムである。アプリケーションプログラム29bにより、図29に示すテーブルT11が管理される。図29は、OS28a上で動作する各プログラムについて各処理単位の動作量を格納するテーブルを示す。

50

## 【0099】

アプリケーションプログラム29bにより、図27、図28に示される関係と、各スレッドコンテキストまたは各プロセスコンテキストに格納される積算値Sの情報に基づいて、図29のテーブルT11が更新される。これは図13、14に示すテーブルT4の更新と同様の処理で行なわれる。これにより、アプリケーションプログラム29aによって識別される処理対象の単位ごとに、処理に要したプロセッサ301の動作量が計測される。アプリケーションプログラム29bにより生成される画面情報は、例えば、対応する処理対象に対して、メールなどの手段により送信される。この画面情報とは、例えば図15や図20に示される、動作量の計測結果を示す画面情報である。これにより、サービスの提供を受けるユーザが、サービスの提供に要したプロセッサ301の動作量を把握可能となる。

## 【0100】

以上の第1の実施形態から第5の実施形態のいずれにおいても、1つの例においては、プロセッサ301のコアが複数設けられる。OSによって、プロセスまたはスレッドの制御に応じて、プロセスまたはスレッドが割り当てられるプロセッサのコアそれぞれに設けられたクロックカウンタから値を読み出すことにより、複数のコアのそれぞれで処理に要した動作量が算出される。

## 【0101】

以上の実施形態に関し、さらに以下の付記を開示する。

## (付記1)

異なる周波数のクロック信号が供給され得るプロセッサを備えるコンピュータに、前記プロセッサへの割り当て制御の対象である処理単位の処理の開始制御に応じて、前記クロック信号のパルスを計数するクロックカウンタから読み出した第1の値と、前記処理の停止制御に応じて前記クロックカウンタから読み出した第2の値との差分を算出する、  
処理を実行させることを特徴とする算出プログラム。

## (付記2)

前記コンピュータに、さらに、  
算出された前記差分に、単位クロック数当たりの所定種類のコスト値を示す係数を乗じて得られるコスト値を算出する、  
処理を実行させることを特徴とする付記1に記載の算出プログラム。

## (付記3)

前記処理の停止制御の実行は、前記処理の処理過程における前記プロセッサの処理状況に応じて判断される、  
ことを特徴とする付記1または付記2に記載の算出プログラム。

## (付記4)

前記コンピュータに、さらに、  
算出した前記差分を、前記処理単位を識別する識別情報と関連付けて記憶部に記憶する、  
処理を実行させることを特徴とする付記1～付記3のいずれか1つに記載の算出プログラム。

## (付記5)

前記コンピュータに、さらに、  
前記処理単位の処理がさらに実行される場合に、さらに実行される処理の開始制御に応じて前記クロックカウンタから読みだされた第3の値と、前記さらに実行される処理の停止制御に応じて前記クロックカウンタから読みだされた第4の値との差分を算出し、  
前記第3の値と前記第4の値とに基づいて算出された前記差分を、前記識別情報と関連付けられて前記記憶部に記憶された前記差分に積算して得られる積算値を、前記識別情報と関連付けて前記記憶部に記憶する、  
処理を実行させることを特徴とする付記4に記載の算出プログラム。

(付記 6)

前記コンピュータに、さらに、

他の処理単位が実行される場合に、前記他の処理単位の処理の開始制御に応じて前記クロックカウンタから読み出した第 5 の値と、前記他の処理単位の前記処理の停止制御に応じて前記クロックカウンタから読み出した第 6 の値との差分を算出し、

前記他の処理単位が前記処理単位と同じプログラムについての処理単位である場合に、前記第 5 の値及び前記第 6 の値に基づいて算出された差分と、前記積算値との和を、前記プログラムを識別するプログラム識別情報に関連付けて前記記憶部に記憶する、

処理を実行させることを特徴とする付記 5 に記載の算出プログラム。

(付記 7)

前記コンピュータに、さらに、

前記プログラムに関する入力制御または出力制御のうち、少なくとも一方の制御を実行させるドライバプログラムの処理単位の処理が実行される場合に、前記ドライバプログラムの処理単位の処理の開始制御に応じて前記クロックカウンタから読み出した第 7 の値と、前記ドライバプログラムの前記処理単位の前記の停止処理の停止制御に応じて前記クロックカウンタから読み出した第 8 の値との差分を算出し、

前記第 7 の値と前記第 8 の値とに基づく前記差分と、前記和との和を、前記プログラム識別情報に関連付けて前記記憶部に記憶する、

処理を実行させることを特徴とする付記 6 に記載の算出プログラム。

(付記 8)

前記処理単位は、マルチスレッド制御の対象となるスレッドであるか、マルチプロセス制御の対象となるプロセスであるか、または前記コンピュータへのアクセス元に関して行なわれる処理の単位である、

ことを特徴とする付記 1 ～ 7 のいずれか 1 つに記載の算出プログラム。

(付記 9)

異なる周波数のクロック信号が供給され得るプロセッサと、前記クロック信号を計数するクロックカウンタと、を含み、

前記プロセッサが、

前記プロセッサへの割り当て制御の対象である処理単位の処理の開始制御に応じて、前記クロックカウンタから読み出した第 1 の値と、前記処理の停止制御に応じて前記クロックカウンタから読み出した第 2 の値との差分を算出する、

処理を実行することを特徴とするコンピュータ。

(付記 10)

前記クロックカウンタが、

前記クロック信号が前記プロセッサに供給されたパルス数を計数する、

ことを特徴とする付記 9 に記載のコンピュータ。

(付記 11)

前記コンピュータは、さらにメモリを備え、

前記プロセッサが、

前記算出した前記差分を、前記処理単位を識別する識別情報に関連付けて前記メモリに記憶する、

処理を実行することを特徴とする付記 9 または付記 10 に記載のコンピュータ。

(付記 11)

異なる周波数のクロック信号が供給され得るプロセッサ、及びクロックカウンタを含むコンピュータが、

前記クロックカウンタにより、前記クロック信号のパルスを計数し、

前記プロセッサへの割り当て制御の対象である処理単位の処理の開始制御に応じて、前記クロックカウンタから読み出した第 1 の値と、前記処理の停止制御に応じて前記クロックカウンタから読み出した第 2 の値との差分を算出する、

ことを実行させることを特徴とする算出方法。

10

20

30

40

50

## 【符号の説明】

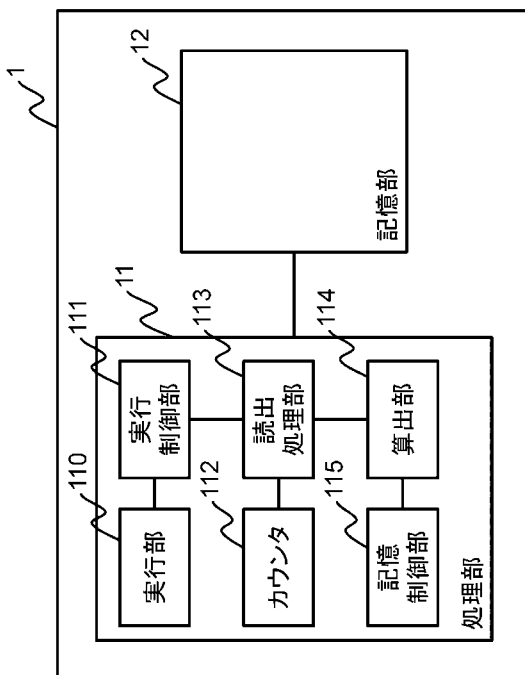
## 【0102】

- 1 コンピュータ  
 11 処理部  
 12 記憶部  
 13 通信部  
 14 処理部  
 15 通信部  
 16 出力部  
 17 入力部  
 110 実行部  
 111 実行制御部  
 112 クロックカウンタ  
 113 読出処理部  
 114 算出部  
 115 記憶制御部  
 21 ハードウェア  
 22a オペレーションシステム  
 22b ハイパーバイザ  
 23a-23c アプリケーションプログラム  
 26a-26c 仮想マシン  
 27a、27b ドライバVM  
 28a、28b オペレーションシステム  
 29a-29d アプリケーションプログラム

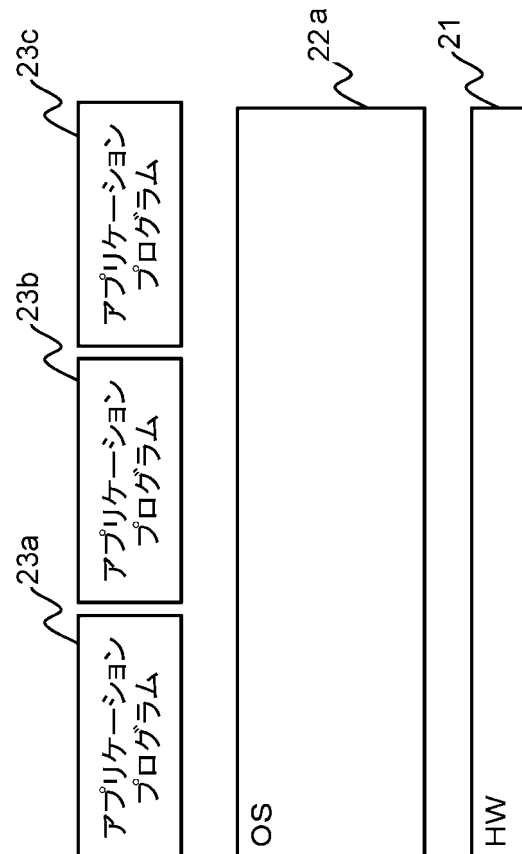
10

20

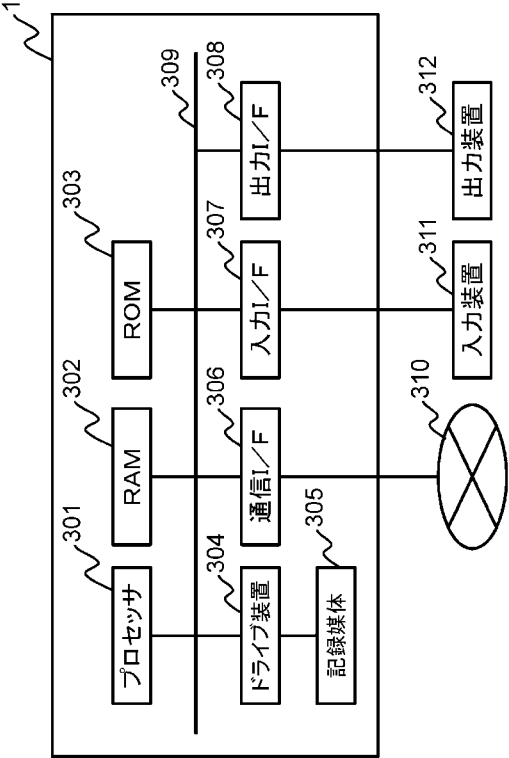
【図1】



【図2】



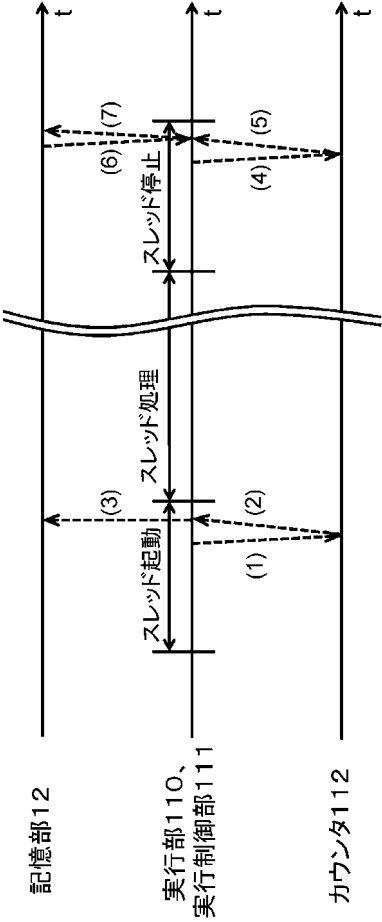
【図 3】



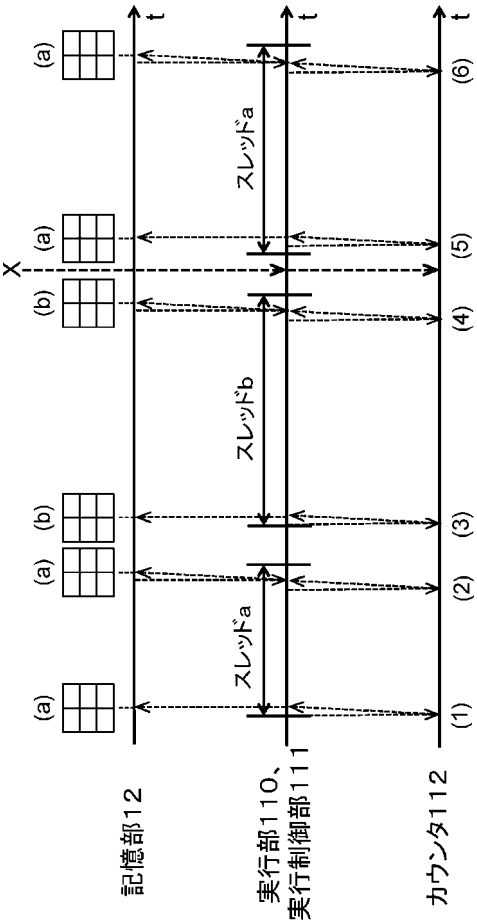
【図 5】

|              |
|--------------|
| スレッドID       |
| 起動時クロック数clk0 |
| クロック数積算値s    |
| 起動時刻t0       |
| 時間積算値st      |
| プログラムカウンタ    |
| スタックポインタ     |

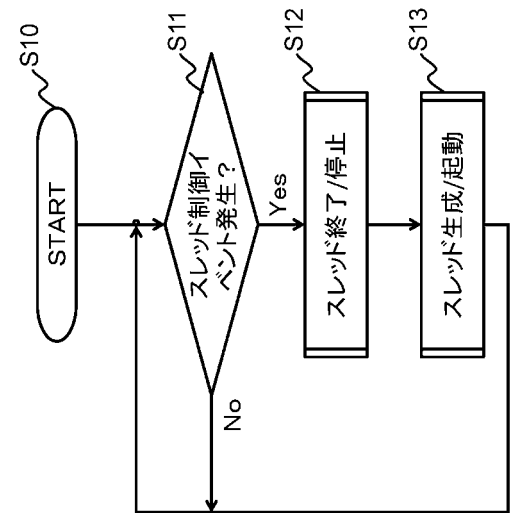
【図 4】



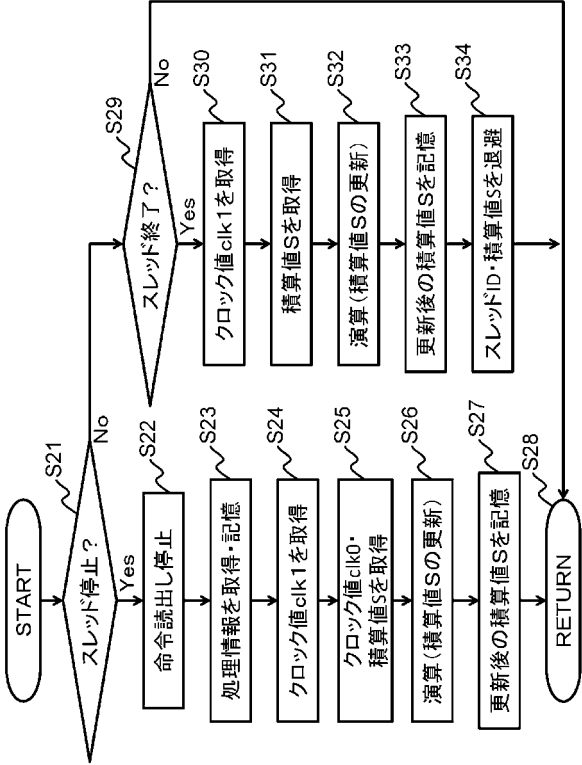
【図 6】



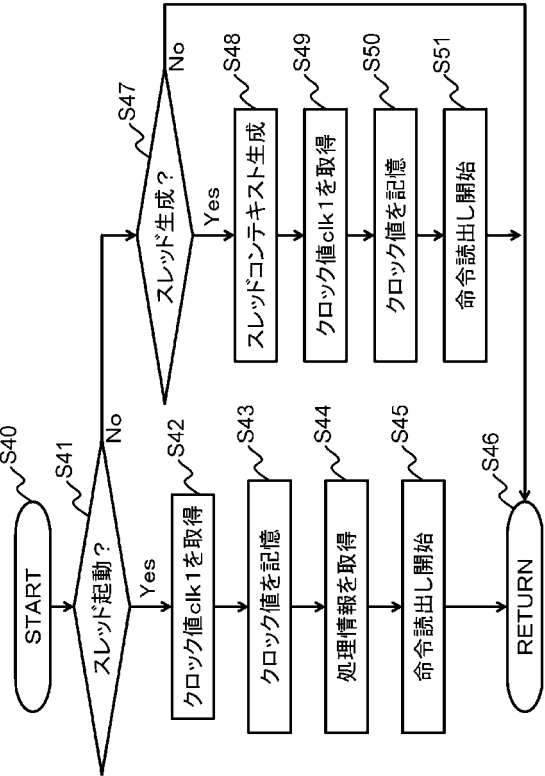
【図 7】



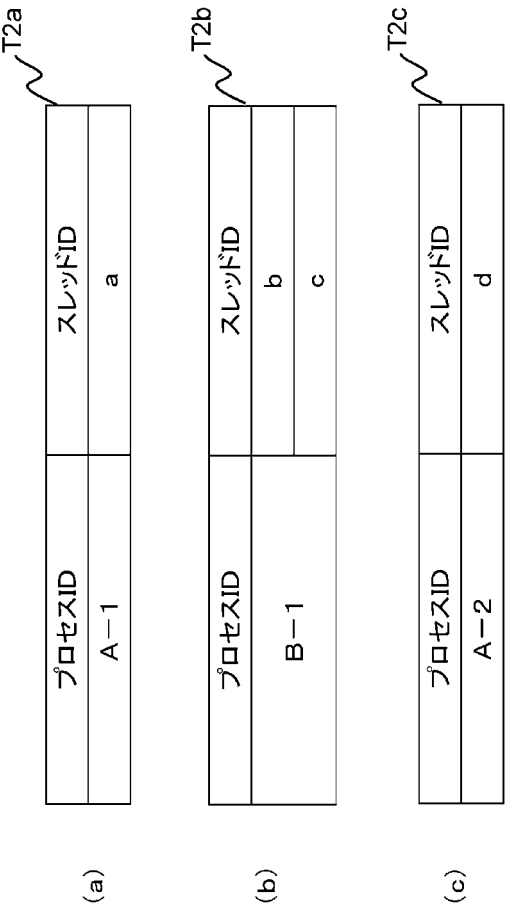
【図 8】



【図 9】



【図 10】



【図 1 1】

T3

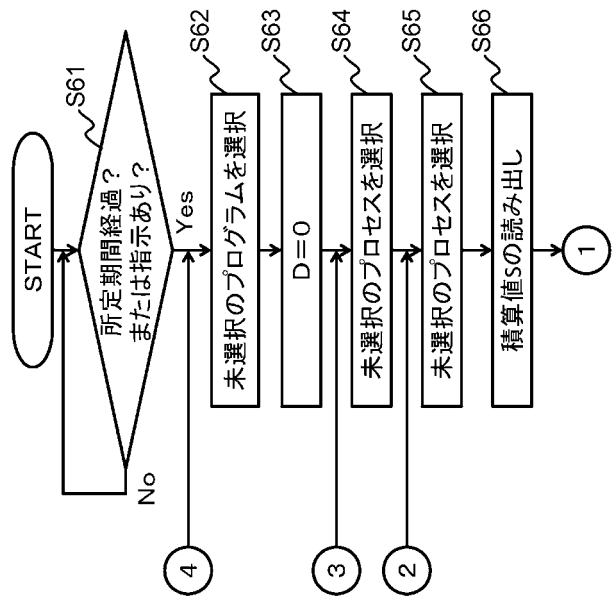
| プログラムID | プロセスID |
|---------|--------|
| A       | A-1    |
|         | A-2    |
| B       | B-1    |
|         | B-2    |
| C       | C-1    |

【図 1 2】

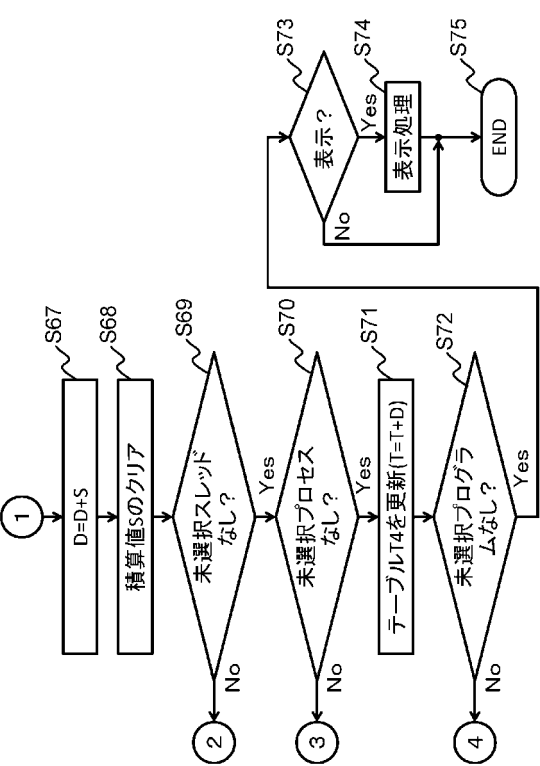
T4

| プログラムID | プログラム動作量(T)   |
|---------|---------------|
| A       | 1234567890123 |
| B       | 456789012345  |
| C       | 67890123      |

【図 1 3】



【図 1 4】



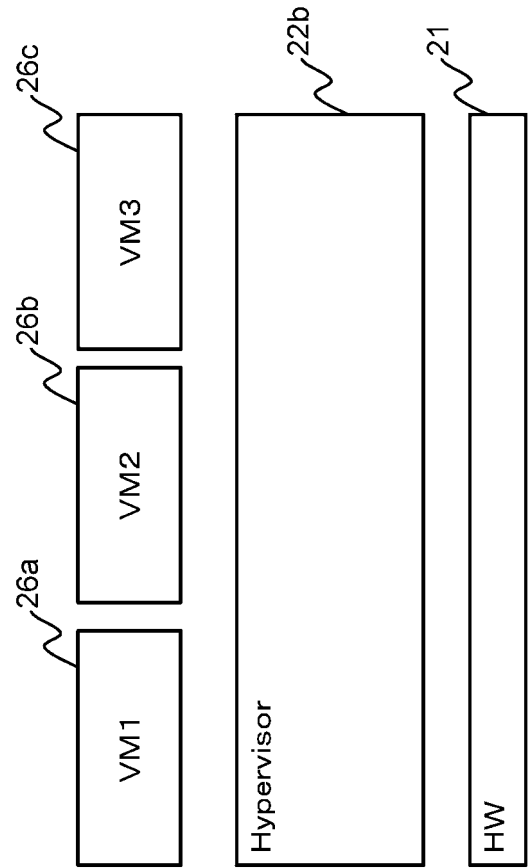


【図 15】

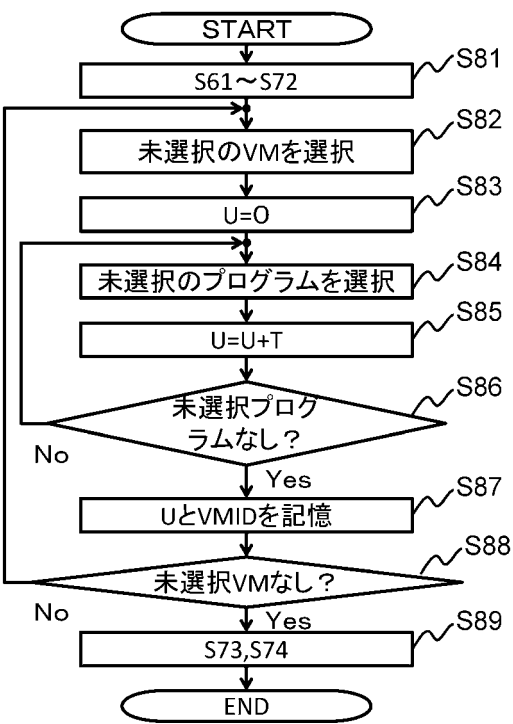
41

| ★アプリケーションの使用状況★ |                            |             |               |                                        |
|-----------------|----------------------------|-------------|---------------|----------------------------------------|
| アプリ<br>ケーション    | クロック数<br>( $\times 10^9$ ) | 消費電力<br>(W) | 発熱量<br>(kcal) | CO2消費<br>( $\times 10^{-6} \text{t}$ ) |
| A               | 1234.6                     |             |               |                                        |
| B               | 456.8                      |             |               |                                        |
| C               | 0.0                        |             |               |                                        |
|                 |                            |             |               |                                        |

【図 16】



【図 17】



【図 18】

T5

| プログラムID | A | B | C | D | E | F |
|---------|---|---|---|---|---|---|
| VMID    | X |   |   | Y |   | Z |

【図 19】

T6

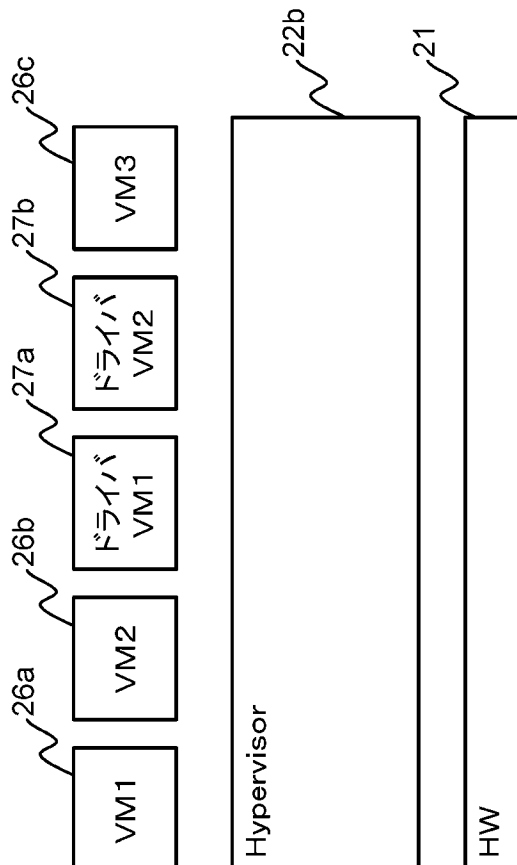
| VMID | VM動作量(U)       |
|------|----------------|
| X    | 12345678901234 |
| Y    | 567890123456   |
| Z    | 78901234       |

【図 20】

42

| ★お客様の使用状況★ |            |                            |             |                              |
|------------|------------|----------------------------|-------------|------------------------------|
| 年／月        | 課金額<br>(円) | クロック数<br>( $\times 10^9$ ) | 消費電力<br>(W) | CO2消費<br>( $t \times 10^5$ ) |
| 2011/11    |            | 12346                      |             |                              |
| 2011/10    |            | 78901                      |             |                              |
| 2011/9     |            | 23456                      |             |                              |
| 2011/8     |            | 78901                      |             |                              |

【図 21】



【図 22】

T7

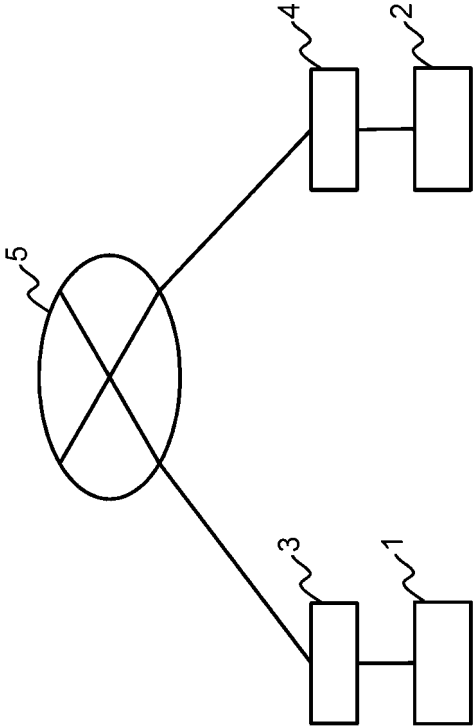
| VMID | プログラムID | VM間の対応関係 |
|------|---------|----------|
| X    | A       | -        |
|      | B       | -        |
|      | C       | -        |
| XX   | XA      | X        |
| Y    | D       | -        |
|      | E       | -        |
| YY   | YA      | Y        |
| Z    | F       | -        |

【図 2 3】

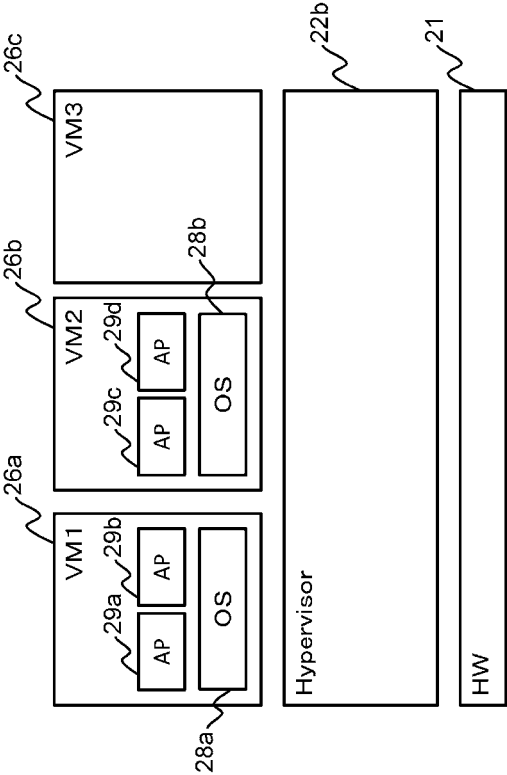
T8

| VMID | VM動作量(U)       |
|------|----------------|
| X    | 12345678901234 |
| XX   | 567890         |
| Y    | 567890123456   |
| YY   | 7890123        |
| Z    | 78901234       |

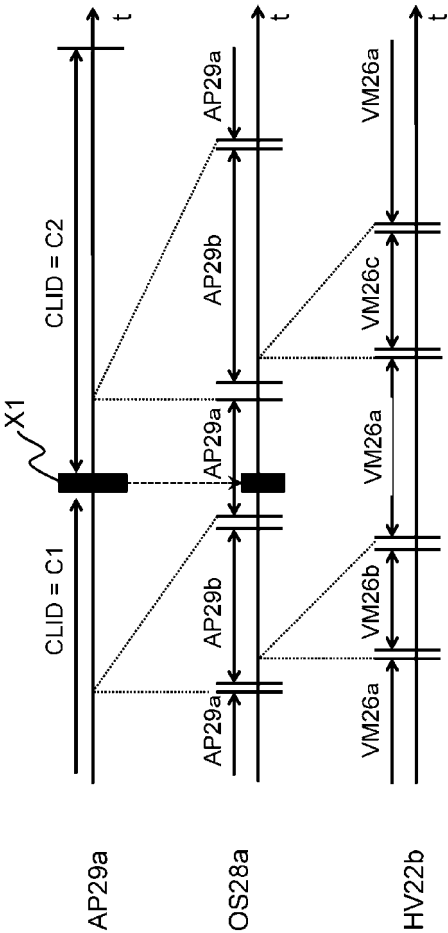
【図 2 4】



【図 2 5】



【図 2 6】



【図 29】

T11

| プログラムID+CLID | プログラム動作量(T)   |
|--------------|---------------|
| A-C1         | 1234567890123 |
| B-C1         | 456789012345  |
| A-C2         | 4561234567890 |

(a)

【図 27】

T9a

| プロセスID+CLID | スレッドID+CLID |
|-------------|-------------|
| A-1-C1      | a-C1        |

T9b

| プロセスID+CLID | スレッドID+CLID |
|-------------|-------------|
| B-1-C1      | b-C1        |
|             | c-C1        |

(b)

T9c

| プロセスID+CLID | スレッドID |
|-------------|--------|
| A-2-C1      | d-C1   |

(c)

【図 28】

T10

| プログラムID+CLID | プロセスID+CLID |
|--------------|-------------|
| A-C1         | A-1-C1      |
|              | A-2-C1      |
| B-C1         | B-1-C1      |
|              | B-2-C1      |
| C-C1         | C-1-C1      |

---

フロントページの続き

(72)発明者 黒羽 法男

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

Fターム(参考) 5B042 GA38 LA23 MA14 MB05 MB06 MC21 MC33