

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号  
特許第7355456号  
(P7355456)

(45)発行日 令和5年10月3日(2023.10.3)

(24)登録日 令和5年9月25日(2023.9.25)

(51)国際特許分類 F I  
G 0 6 F 9/48 (2006.01) G 0 6 F 9/48 3 0 0 E  
G 0 6 F 9/48 3 0 0 J

請求項の数 9 (全19頁)

(21)出願番号	特願2019-222164(P2019-222164)	(73)特許権者	000232092 N E Cソリューションイノベータ株式会 社 東京都江東区新木場一丁目18番7号
(22)出願日	令和1年12月9日(2019.12.9)	(74)代理人	100124811 弁理士 馬場 資博
(65)公開番号	特開2021-92904(P2021-92904A)	(74)代理人	100088959 弁理士 境 廣巳
(43)公開日	令和3年6月17日(2021.6.17)	(74)代理人	100097157 弁理士 桂木 雄二
審査請求日	令和4年11月2日(2022.11.2)	(74)代理人	100187724 弁理士 唐鎌 睦
		(72)発明者	三角 愛子 東京都江東区新木場一丁目18番7号 N E Cソリューションイノベータ株式会 最終頁に続く

(54)【発明の名称】 C P Uリソース管理装置

(57)【特許請求の範囲】

【請求項1】

プロセスを格納する記憶装置と、前記プロセスが実行されるプロセッサと、を備えたC P Uリソース管理装置であって、

前記プロセッサは、

インターバルを開始すると、C P U使用時間が最大C P U使用時間以下である条件を満たすワークロードクラスに属するプロセスにC P Uリソースを割り当てて実行し、前記条件を満たすワークロードクラスに属するプロセスが存在しなくなった時点で前記インターバルを終了し、前記最大C P U使用時間に達したことが原因で行われたC P Uリソース割り当ての打ち切り回数と前記インターバルの開始から終了までの時間である総割り当て時間とを記録する実行制御手段と、

前記総割り当て時間がベースインターバル時間より短く、且つ、前記C P Uリソース割り当ての打ち切り回数が1以上の場合、次回のインターバル時間を前回のインターバル時間より長い時間に決定し、且つ、前記次回のインターバル時間とワークロードクラス毎のC P U使用時間の配分比とに基づいて次回のワークロードクラス毎の最大C P U使用時間を決定する調整手段と、

を備えるC P Uリソース管理装置。

【請求項2】

前記調整手段は、前記総割り当て時間がベースインターバル時間より長い場合、次回のインターバル時間を前回のインターバル時間より短い時間に決定し、且つ、前記次回のイ

ンターバル時間とワークロードクラス毎のCPU使用時間の配分比とに基づいて次回のワークロードクラス毎の最大CPU使用時間を決定する、  
請求項1に記載のCPUリソース管理装置。

【請求項3】

前記調整手段は、前記総割り当て時間に対する前記ベースインターバル時間の比を前回のインターバル時間に乗じることにより、次回のインターバル時間を計算する、  
請求項1または2に記載のCPUリソース管理装置。

【請求項4】

前記調整手段は、前記総割り当て時間に対する前記ベースインターバル時間の比を前回のインターバル時間に乗じて得られる値を計算し、ベースインターバル時間に対する前記CPUリソース割り当ての打ち切りによるプロセス切替にかかった総時間の割合が閾値を超えていれば、前記値に前記割合を乗じて得られる値をさらに前記閾値で除算した値を、  
次回のインターバル時間として計算する、  
請求項1または2に記載のCPUリソース管理装置。

10

【請求項5】

プロセスを格納する記憶装置と、前記プロセスが実行されるプロセッサと、を備えたコンピュータが実行するCPUリソース管理方法であって、

インターバルを開始すると、CPU使用時間が最大CPU使用時間以下である条件を満たすワークロードクラスに属するプロセスにCPUリソースを割り当てて実行し、前記条件を満たすワークロードクラスに属するプロセスが存在しなくなった時点で前記インターバルを終了し、前記最大CPU使用時間に達したことが原因で行われたCPUリソース割り当ての打ち切り回数と前記インターバルの開始から終了までの時間である総割り当て時間とを記録し、

20

前記総割り当て時間がベースインターバル時間より短く、且つ、前記CPUリソース割り当ての打ち切り回数が1以上の場合、次回のインターバル時間を前回のインターバル時間より長い時間に決定し、且つ、前記次回のインターバル時間とワークロードクラス毎のCPU使用時間の配分比とに基づいて次回のワークロードクラス毎の最大CPU使用時間を決定する、

CPUリソース管理方法。

【請求項6】

前記総割り当て時間がベースインターバル時間より長い場合、次回のインターバル時間を前回のインターバル時間より短い時間に決定し、且つ、前記次回のインターバル時間とワークロードクラス毎のCPU使用時間の配分比とに基づいて次回のワークロードクラス毎の最大CPU使用時間を決定する、

請求項5に記載のCPUリソース管理方法。

30

【請求項7】

前記総割り当て時間に対する前記ベースインターバル時間の比を前回のインターバル時間に乗じることにより、次回のインターバル時間を計算する、

請求項5または6に記載のCPUリソース管理方法。

【請求項8】

前記総割り当て時間に対する前記ベースインターバル時間の比を前回のインターバル時間に乗じて得られる値を計算し、ベースインターバル時間に対する前記CPUリソース割り当ての打ち切りによるプロセス切替にかかった総時間の割合が閾値を超えていれば、前記値に前記割合を乗じて得られる値をさらに前記閾値で除算した値を、次回のインターバル時間として計算する、

請求項5または6に記載のCPUリソース管理方法。

40

【請求項9】

プロセスを格納する記憶装置と、前記プロセスが実行されるプロセッサと、を備えたコンピュータに、

インターバルを開始すると、CPU使用時間が最大CPU使用時間以下である条件を満た

50

たすワークロードクラスに属するプロセスにCPUリソースを割り当てて実行し、前記条件を満たすワークロードクラスに属するプロセスが存在しなくなった時点で前記インターバルを終了し、前記最大CPU使用時間に達したことが原因で行われたCPUリソース割り当ての打ち切り回数と前記インターバルの開始から終了までの時間である総割り当て時間とを記録する処理と、

前記総割り当て時間がベースインターバル時間より短く、且つ、前記CPUリソース割り当ての打ち切り回数が1以上の場合、次のインターバル時間を前回のインターバル時間より長い時間に決定し、且つ、前記次のインターバル時間とワークロードクラス毎のCPU使用時間の配分比とに基づいて次のワークロードクラス毎の最大CPU使用時間を決定する処理と、

を行わせるためのプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、CPUリソース管理装置、CPUリソース管理方法、および、プログラムに関する。

【背景技術】

【0002】

近年のコンピュータシステムでは、システム管理者が、プロセスが属するクラスであるワークロードクラスを設定し、ワークロードクラス毎に利用可能なCPUリソースを割り当てること行われている。このようなコンピュータシステムでは、各ワークロードクラスに対するリソース配分を適切に行わないと、システム性能が大きく低下し、システムの運用に支障をきたすことがある。そのため、これまでに、各ワークロードクラスに対するCPUリソース配分を工夫して行うための技術が開示されている。

【0003】

例えば、特許文献1には、各ワークロードクラスに割り当てるCPUリソースの配分比に従い、一定の時間であるインターバル毎に、各ワークロードクラスにCPUリソースを割り当てる最大CPU使用時間を設定し、この最大CPU使用時間に従い、各ワークロードクラスに属するプロセスにCPUリソースを割り当てる技術が記載されている。例えば、a、b、cの3つのワークロードクラスがあり、そのCPUリソース配分比を4:3:3とし、インターバル時間を100msとすると、1回のインターバルでは、ワークロードクラスa、b、cに属する各プロセスに、それぞれ最大40ms、30ms、30msのCPU使用時間が割り当てられることになる。

【先行技術文献】

【特許文献】

【0004】

【文献】特許第4427802号

【発明の概要】

【発明が解決しようとする課題】

【0005】

ところで、インターバル内で、割り当てられた最大CPU使用時間を必要としない低負荷なワークロードクラスが存在する場合、余剰CPU時間分だけ、インターバルを早く終了し、次のインターバルを開始することが望ましい。例えば、ワークロードクラスaは高負荷な状態にあり、割り当てられた最大CPU使用時間以上にCPUリソースを必要とするが、ワークロードクラスb、cは低負荷な状態にあり、割り当てられた最大CPU使用時間を必要としないものとする。このとき、ワークロードクラスb、cが割り当てられた最大CPU使用時間の30msのうち例えば15msだけしか使用しなかった場合、30msの余剰CPU時間分だけ、インターバルを早く終了し、次のインターバルを開始する。この結果、前回のインターバルは実質70msに短縮される。次のインターバルでも同様の状況が生じると、次のインターバルも実質70msに短縮される。

10

20

30

40

50

## 【0006】

このように割り当てられた最大CPU使用時間を必要としない低負荷なワークロードクラスの余剰CPU時間分だけインターバルを短縮することにより、割り当てられた最大CPU使用時間以上にCPUリソースを必要とする高負荷なワークロードクラスのターンアラウンド時間を改善することができる。しかしながら、インターバルを短縮すればするほどに単位時間当たりのプロセス切り替え回数が増大するため、スループットが低下する。

## 【0007】

本発明の目的は、上述した課題を解決するCPUリソース管理装置を提供することにある。

## 【課題を解決するための手段】

## 【0008】

本発明の一形態に係るCPUリソース管理装置は、  
プロセスを格納する記憶装置と、前記プロセスが実行されるプロセッサと、を備えたCPUリソース管理装置であって、

前記プロセッサは、  
インターバルを開始すると、CPU使用時間が最大CPU使用時間以下である条件を満たすワークロードクラスに属するプロセスにCPUリソースを割り当てて実行し、前記条件を満たすワークロードクラスに属するプロセスが存在しなくなった時点で前記インターバルを終了し、前記最大CPU使用時間に達したことが原因で行われたCPUリソース割り当ての打ち切り回数と前記インターバルの開始から終了までの時間である総割り当て時間とを記録する実行制御手段と、

前記総割り当て時間がベースインターバル時間より短く、且つ、前記CPUリソース割り当ての打ち切り回数が1以上の場合、次回のインターバル時間を前回のインターバル時間より長い時間に決定し、且つ、前記次回のインターバル時間とワークロードクラス毎のCPU使用時間の配分比とに基づいて次回のワークロードクラス毎の最大CPU使用時間を決定する調整手段と、  
を備えるように構成されている。

## 【0009】

また本発明の他の形態に係るCPUリソース管理方法は、  
プロセスを格納する記憶装置と、前記プロセスが実行されるプロセッサと、を備えたコンピュータが実行するCPUリソース管理方法であって、

インターバルを開始すると、CPU使用時間が最大CPU使用時間以下である条件を満たすワークロードクラスに属するプロセスにCPUリソースを割り当てて実行し、前記条件を満たすワークロードクラスに属するプロセスが存在しなくなった時点で前記インターバルを終了し、前記最大CPU使用時間に達したことが原因で行われたCPUリソース割り当ての打ち切り回数と前記インターバルの開始から終了までの時間である総割り当て時間とを記録し、

前記総割り当て時間がベースインターバル時間より短く、且つ、前記CPUリソース割り当ての打ち切り回数が1以上の場合、次回のインターバル時間を前回のインターバル時間より長い時間に決定し、且つ、前記次回のインターバル時間とワークロードクラス毎のCPU使用時間の配分比とに基づいて次回のワークロードクラス毎の最大CPU使用時間を決定する、  
ように構成されている。

## 【0010】

また本発明の他の形態に係るプログラムは、  
プロセスを格納する記憶装置と、前記プロセスが実行されるプロセッサと、を備えたコンピュータに、

インターバルを開始すると、CPU使用時間が最大CPU使用時間以下である条件を満たすワークロードクラスに属するプロセスにCPUリソースを割り当てて実行し、前記条件を満たすワークロードクラスに属するプロセスが存在しなくなった時点で前記インター

10

20

30

40

50

バルを終了し、前記最大CPU使用時間に達したことが原因で行われたCPUリソース割り当ての打ち切り回数と前記インターバルの開始から終了までの時間である総割り当て時間とを記録する処理と、

前記総割り当て時間がベースインターバル時間より短く、且つ、前記CPUリソース割り当ての打ち切り回数が1以上の場合、次のインターバル時間を前回のインターバル時間より長い時間に決定し、且つ、前記次のインターバル時間とワークロードクラス毎のCPU使用時間の配分比とに基づいて次のワークロードクラス毎の最大CPU使用時間を決定する処理と、

を行わせるように構成されている。

【発明の効果】

【0011】

本発明は上述したような構成を有することにより、コンピュータシステムのスループットを改善することができる。

【図面の簡単な説明】

【0012】

【図1】本発明の第1の実施形態に係るCPUリソース管理装置のブロック図である。

【図2】本発明の第1の実施形態に係るCPUリソース管理装置におけるワークロード定義の構成例を示す図である。

【図3】本発明の第1の実施形態に係るCPUリソース管理装置における実行可能プロセスキューの構成例を示す図である。

【図4】本発明の第1の実施形態に係るCPUリソース管理装置における次のインターバル時間見直し処理(1)で使用する判定ルールの構成例を示す図である。

【図5】本発明の第1の実施形態に係るCPUリソース管理装置における次のインターバル時間見直し処理(1)で使用する計算式を示す図である。

【図6】本発明の第1の実施形態に係るCPUリソース管理装置における次のインターバル時間見直し処理(2)で使用する条件式を示す図である。

【図7】本発明の第1の実施形態に係るCPUリソース管理装置における次のインターバル時間見直し処理(2)で使用する計算式を示す図である。

【図8】本発明の第1の実施形態に係るCPUリソース管理装置の動作を表すフローチャートである。

【図9】本発明の第1の実施形態に係るCPUリソース管理装置の作用効果の説明図である。

【図10】本発明の第2の実施形態に係るCPUリソース管理装置のブロック図である。

【図11】本発明の第3の実施形態に係るCPUリソース管理装置のブロック図である。

【発明を実施するための形態】

【0013】

次に本発明の実施の形態について図面を参照して詳細に説明する。

[第1の実施形態]

図1は本発明の第1の実施形態に係るCPUリソース管理装置100のブロック図である。図1を参照すると、CPUリソース管理装置100は、記憶装置110と、この記憶装置110に接続されたCPU120とから構成されている。

【0014】

記憶装置110は、ハードディスクやメモリなどから構成され、CPU120で実行される各種処理に必要な処理情報およびプログラム114を記憶するように構成されている。プログラム114は、CPU120に読み込まれて実行されることにより各種処理部を実現するプログラムであり、図示しない通信インターフェース部などのデータ入出力機能を介して外部装置(図示せず)や記憶媒体(図示せず)から予め読み込まれて記憶装置110に保存される。記憶装置110に記憶される処理情報には、ワークロード定義111、実行可能プロセスキュー112、および、ディスパッチング情報113がある。

【0015】

10

20

30

40

50

ワークロード定義 1 1 1 は、プロセス（もしくはジョブ）が属するクラス（ワークロードクラス）を定義するクラス定義と、各ワークロードクラスに割り当てる CPU リソースの配分比（または配分率）を定義する配分定義とを表す情報である。ワークロード定義 1 1 1 は、主に、システム管理者により設定される。図 2 は、ワークロード定義 1 1 1 の構成例である。このワークロード定義 1 1 1 のクラス定義では、それぞれ a、b、c という名前の 3 つのワークロードクラスが存在することが定義されている。ワークロードクラスは、主に、業務単位に分けられる。また、このワークロード定義 1 1 1 の配分定義では、ワークロードクラス a、b、c の CPU リソースの配分比が 4 : 3 : 3 であることが定義されている。

【 0 0 1 6 】

実行可能プロセスキュー 1 1 2 は、実行可能なプロセスがキューイングされる記憶手段である。図 3 は、実行可能プロセスキュー 1 1 2 の構成例である。この実行可能プロセスキュー 1 1 2 では、キューイングされたプロセスのプロセス番号および所属するワークロードクラスの情報が記憶されている。実行可能プロセスキュー 1 1 2 には、プロセス番号および所属するワークロードクラスの情報以外に、各プロセスの実行優先度などの情報が記憶されていてもよい。また、各プロセスが所属するワークロードクラスは、実行可能プロセスキュー 1 1 2 以外の箇所で記憶するようにしてもよい。

【 0 0 1 7 】

ディスパッチング情報 1 1 3 は、プロセスに CPU リソースを割り当てる制御に使用する情報である。図 1 を参照すると、ディスパッチング情報 1 1 3 は、ベースインターバル時間 1 3 1、プロセス切替平均時間 1 3 2、閾値 1 3 3、次回のインターバル時間 1 3 4、次回のワークロードクラス毎の最大 CPU 使用時間 1 3 5、前回のインターバル時間 1 3 6、前回の総割り当て時間 1 3 7、および、前回の CPU リソース割り当て打ち切り回数 1 3 8 を含んで構成されている。

【 0 0 1 8 】

ベースインターバル時間 1 3 1 は、基本となるインターバル時間であり、固定値である。例えば、ベースインターバル時間 1 3 1 は、1 0 0 m s としてよい。

【 0 0 1 9 】

プロセス切替平均時間 1 3 2 は、CPU 上で実行するプロセスを別のプロセスに切り替えるために必要な時間の平均値である。プロセス切替平均時間 1 3 2 は、1 回のプロセス切替にかかる平均時間である。

【 0 0 2 0 】

閾値 1 3 3 は、後述する次回のインターバル時間見直し処理(2)の判定で使用する閾値である。

【 0 0 2 1 】

次回のインターバル時間 1 3 4 は、次回のインターバルで使用されるインターバル時間である。この次回のインターバル時間 1 3 4 は可変値である。

【 0 0 2 2 】

次回のワークロードクラス毎の最大 CPU 使用時間 1 3 5 は、次回のインターバルにおけるワークロードクラス毎の最大 CPU 使用時間である。最大 CPU 使用時間は、配分時間とも称される。

【 0 0 2 3 】

前回のインターバル時間 1 3 6 は、前回のインターバルで使用されたインターバル時間である。

【 0 0 2 4 】

前回の総割り当て時間 1 3 7 は、前回のインターバルの開始から終了までの時間である。

【 0 0 2 5 】

前回の CPU リソース割り当て打ち切り回数 1 3 8 は、前回のインターバルの開始から終了までに最大 CPU 使用時間に達したことが原因で行われた CPU リソース割り当ての打ち切り回数である。

10

20

30

40

50

## 【 0 0 2 6 】

C P U 1 2 0 は、演算処理装置とその周辺回路とから構成され、記憶装置 1 1 0 からプログラム 1 1 4 を読み込んで実行することにより、ハードウェアとプログラム 1 1 4 とを協働させて各種処理部を実現するように構成されている。C P U 1 2 0 で実現される処理部には、初期化手段 1 2 1 と実行制御手段 1 2 2 と調整手段 1 2 3 とがある。

## 【 0 0 2 7 】

初期化手段 1 2 1 は、ワークロード定義 1 1 1 およびディスパッチング情報 1 1 3 を初期化するように構成されている。具体的には、初期化手段 1 2 1 は、図示しない通信インターフェース部などのデータ入出力機能を介してシステム管理者からワークロード定義 1 1 1 を入力し、記憶装置 1 1 0 に記憶する。また初期化手段 1 2 1 は、ベースインターバル時間 1 3 1 に予め定められた固定時間（例えば 1 0 0 m s ）を設定する。また初期化手段 1 2 1 は、プロセス切替平均時間 1 3 2 に事前の測定などによって決定した 1 回のプロセス切替にかかる平均時間（例えば 1 0 μ s ）を設定する。また初期化手段 1 2 1 は、閾値 1 3 3 に予め定められた値（例えば 5 % ）を設定する。また初期化手段 1 2 1 は、次のインターバル時間 1 3 4 にベースインターバル時間 1 3 1 と同じ時間を設定する。また初期化手段 1 2 1 は、次のワークロードクラス毎の最大 C P U 使用時間 1 3 5 に、次のインターバル時間 1 3 4 に設定したインターバル時間をワークロード定義 1 1 1 に記載されるワークロードクラス毎の C P U 使用時間の配分比に従って配分して計算したワークロードクラス毎の最大 C P U 使用時間を設定する。また初期化手段 1 2 1 は、前回の C P U リソース割り当て打ち切り回数 1 3 8 に初期値（例えば 0 ）を設定する。但し、初期化手段 1 2 1 は、前回のインターバル時間 1 3 6 、前回の総割り当て時間 1 3 7 、および、前回の C P U リソース割り当て打ち切り回数 1 3 8 に前回の履歴が残されている場合、それらを初期化しないようにしてもよい。

## 【 0 0 2 8 】

実行制御手段 1 2 2 は、プロセスに C P U リソースを割り当てて実行するように構成されている。具体的には、実行制御手段 1 2 2 は、ディスパッチング情報 1 1 3 を記憶装置 1 1 0 から読み出し、そのディスパッチング情報 1 1 3 に従って、実行可能プロセスキュー 1 1 2 にキューイングされている各ワークロードクラスのプロセスを実行するインターバルを開始する。また実行制御手段 1 2 2 は、インターバルを開始すると、内部変数として有するワークロードクラス毎の C P U 使用時間と C P U リソース割り当て打ち切り回数をそれぞれ零にリセットし、次のインターバル時間 1 3 4 をセットしたタイマの減算を開始する。また実行制御手段 1 2 2 は、インターバル開始後の C P U 使用時間が、次のワークロードクラス毎の最大 C P U 使用時間 1 3 5 に設定されている最大 C P U 使用時間以下であるワークロードクラスに属するプロセスを実行可能プロセスキュー 1 1 2 から取り出して C P U 1 2 0 を割り当てて実行する。また実行制御手段 1 2 2 は、プロセスの実行時間に応じて、そのプロセスが属するワークロードクラスの C P U 使用時間を加算する。また実行制御手段 1 2 2 は、プロセスの実行中に、そのプロセスが属するワークロードクラスの C P U 使用時間がその最大 C P U 使用時間 1 3 5 を超えると、そのプロセスへの C P U リソースの割り当てを打ち切り、そのプロセスを実行可能プロセスキュー 1 1 2 に格納する。その際、実行制御手段 1 2 2 は、C P U リソース割り当て打ち切り回数をインクリメントする。また、実行中のプロセスが自ら C P U を離す事象（入出力要求など）が発生した場合、実行制御手段 1 2 2 は、そのプロセスへの C P U リソースの割り当てを打ち切り、そのプロセスを図示しない実行待ちキューなどに格納する。このとき実行制御手段 1 2 2 は、C P U リソース割り当ての打ち切りを行った原因が最大 C P U 使用時間によるものでないため、C P U リソース割り当て打ち切り回数はインクリメントしない。

## 【 0 0 2 9 】

一方、実行制御手段 1 2 2 は、インターバル開始後の C P U 使用時間が次のワークロードクラス毎の最大 C P U 使用時間 1 3 5 に設定されている最大 C P U 使用時間以下であるワークロードクラスに属するプロセスが、実行可能プロセスキュー 1 1 2 に存在しない場合、次のインターバル時間 1 3 4 をセットしたタイマの減算を停止させ、今回のイン

10

20

30

40

50

ターバルを終了する。また実行制御手段 1 2 2 は、インターバル開始後の CPU 使用時間が次のワークロードクラス毎の最大 CPU 使用時間 1 3 5 に設定されている最大 CPU 使用時間以下であるワークロードクラスに属するプロセスが、実行可能プロセスキュー 1 1 2 に存在していても、次のインターバル時間 1 3 4 をセットしたタイマの値が零になってタイムアウトすると、今回のインターバルを終了する。

#### 【 0 0 3 0 】

また実行制御手段 1 2 2 は、インターバルを終了すると、次のような更新処理をディスプレイパッチング情報 1 1 3 に対して行った後、更新後のディスプレイパッチング情報 1 1 3 を記憶装置 1 1 0 に書き戻す。まず実行制御手段 1 2 2 は、前記のインターバル時間 1 3 6 に、次のインターバル時間 1 3 4 に設定されていた時間を記憶する。また実行制御手段 1 2 2 は、次のインターバル時間 1 3 4 から停止またはタイムアップしたタイマの値を減算することによって、今回のインターバルの開始から終了までの時間を総割り当て時間として計算し、前回の総割り当て時間 1 3 7 に記憶する。例えば、次のインターバル時間 1 3 4 が例えば 1 0 0 m s の場合、インターバルを開始するとタイマは 1 0 0 m s から減算を開始するため、例えばタイマ値が 3 0 m s の時点でインターバルを終了すると、 $1 0 0 \text{ m s} - 3 0 \text{ m s} = 7 0 \text{ m s}$  が総割り当て時間となる。また実行制御手段 1 2 2 は、インターバル開始から終了までにカウントした CPU リソース割り当て打ち切り回数を前回の CPU リソース割り当て打ち切り回数 1 3 8 に記憶する。

#### 【 0 0 3 1 】

調整手段 1 2 3 は、実行制御手段 1 2 2 による前回のインターバルにおける各ワークロードクラスのプロセスの実行状況に基づいて、次のインターバル時間および次のワークロードクラス毎の最大 CPU 使用時間を調整するように構成されている。具体的には、調整手段 1 2 3 は、実行制御手段 1 2 2 によって更新されたディスプレイパッチング情報 1 1 3 を記憶装置 1 1 0 から読み出し、次のインターバル時間を以下のようにして見直す。

#### 【 0 0 3 2 】

< 次のインターバル時間見直し処理 (1) >

まず調整手段 1 2 3 は、前回の総割り当て時間 1 3 7 とベースインターバル時間 1 3 1 とを比較し、前回の総割り当て時間 1 3 7 が、ベースインターバル時間 1 3 1 より長いか、同じか、短いかを判断する。また調整手段 1 2 3 は、前回の CPU リソース割り当て打ち切り回数 1 3 8 が 1 以上か否か、すなわち、CPU リソースの割り当てが打ち切られたワークロードクラスが存在したか否かを判断する。次に調整手段 1 2 3 は、上記 2 つの判断の結果と予め定められた判定ルールとに基づいて、次のインターバル時間を前回のインターバル時間 1 3 6 から変更するか、変更しないかを判断する。

#### 【 0 0 3 3 】

図 4 は、判定ルールの構成例である。この判定ルールでは、以下の 2 つのケースの場合、次のインターバル時間を前回のインターバル時間から変更し、それ以外のケースの場合、次のインターバル時間を変更せずに前回のインターバル時間と同じにする。

< ケース 1 >

CPU リソースの割り当てが打ち切られたワークロードクラスが存在し、且つ、前回の総割り当て時間がベースインターバルより短い。

< ケース 2 >

前回の総割り当て時間がベースインターバルより長い。CPU リソースの割り当てが打ち切られたワークロードクラスが存在するか否かは問わない。

#### 【 0 0 3 4 】

調整手段 1 2 3 は、上記判定において、次のインターバル時間を前回のインターバル時間から変更すると判定した場合、ケース 1 の場合は、次のインターバル時間を前回のインターバル時間よりも長い時間に変更し、ケース 2 の場合は、次のインターバル時間を前回のインターバル時間よりも短い時間に変更する。具体的には、調整手段 1 2 3 は、図 5 に示す計算式 ( 1 ) に従って、次のインターバル時間を計算する。即ち、調整手段 1 2 3 は、前回の総割り当て時間 1 3 7 に対するベースインターバル時間 1 3 1 の比を前

10

20

30

40

50

回のインターバル時間 1 3 6 に乗じて得られる時間を、次回のインターバル時間とする。

【 0 0 3 5 】

< 次回のインターバル時間見直し処理(2) >

調整手段 1 2 3 は、上記次回のインターバル時間見直し処理(1)に引き続き、次回のインターバル時間見直し処理(2)を実行する。次回のインターバル時間見直し処理(2)では、先ず調整手段 1 2 3 は、図 6 に示す条件式 ( 2 ) が成立するか否かを判断する。即ち、調整手段 1 2 3 は、ベースインターバル時間に対する、最大 CPU 使用時間に達したことが原因で行われたプロセス切替による総時間 ( プロセス切替平均時間 × n ) の割合が、閾値以上か否かを判断する。ここで、条件式 2 および後述する計算式 3 において、n は前回の CPU リソース割り当て打ち切り回数 1 3 8 である。

10

【 0 0 3 6 】

次に調整手段 1 2 3 は、条件式 2 が成立しない場合、次回のインターバル時間見直し処理(1)で決定した次回のインターバル時間を、最終的な次回のインターバル時間に決定する。一方、調整手段 1 2 3 は、条件式 2 が成立する場合、次回のインターバル時間見直し処理(1)で決定した次回のインターバル時間が、更に長くなるように調整する。具体的には、調整手段 1 2 3 は、図 7 に示す計算式 ( 3 ) に従って、次回のインターバル時間を計算する。即ち、調整手段 1 2 3 は、次回のインターバル時間見直し処理(1)で決定した次回のインターバル時間に、ベースインターバル時間に対する最大 CPU 使用時間に達したことが原因で行われたプロセス切替による総時間の割合を乗じた値に、更に閾値 1 3 3 で除算した値を、最終的な次回のインターバル時間とする。

20

【 0 0 3 7 】

調整手段 1 2 3 は、最終的な次回のインターバル時間を決定すると、その最終的な次回のインターバル時間を、ディスパッチング情報 1 1 3 の次回のインターバル時間 1 3 4 に設定する。また調整手段 1 2 3 は、次回のインターバル時間 1 3 4 に設定したインターバル時間をワークロード定義 1 1 1 に記載されるワークロードクラス毎の CPU 使用時間の配分比に従って配分して計算したワークロードクラス毎の最大 CPU 使用時間 1 3 5 をディスパッチング情報 1 1 3 に設定する。そして、調整手段 1 2 3 は、更新後のディスパッチング情報 1 1 3 を記憶装置 1 1 0 に書き戻す。

【 0 0 3 8 】

図 8 は、本実施形態に係る CPU リソース管理装置 1 0 0 の動作を示すフローチャートである。以下、図 8 を参照して、本実施形態に係る CPU リソース管理装置 1 0 0 の動作について説明する。

30

【 0 0 3 9 】

CPU リソース管理装置 1 0 0 は、起動すると、先ず初期化手段 1 2 1 が動作して初期化を行い ( ステップ S 1 ) 、その後実行制御手段 1 2 2 が 1 つのインターバルによってプロセスの実行制御を行う ( ステップ S 2 ) 。また実行制御手段 1 2 2 によって 1 つのインターバルが終了する毎に、調整手段 1 2 3 は、可及的速やかに、前回のインターバルの状況に基づいて次回のインターバル時間およびワークロードクラス毎の最大 CPU 使用時間を調整する ( ステップ S 3 ) 。そして再び、実行制御手段 1 2 2 は、調整手段 1 2 3 の処理が終了すると、可及的速やかに、調整後のインターバル時間およびワークロードクラス毎の CPU 使用時間に基づくインターバルを開始してプロセスの実行制御を行う ( ステップ S 2 ) 。このステップ S 2 、 S 3 のループは、システムの運用が停止されるまで継続して実行される。

40

【 0 0 4 0 】

ステップ S 1 の初期化では、初期化手段 1 2 1 は、前述したように記憶装置 1 1 0 上のワークロード定義 1 1 1 およびディスパッチング情報 1 1 3 を初期化する。

【 0 0 4 1 】

ステップ S 2 の実行制御では、実行制御手段 1 2 2 は、ディスパッチング情報 1 1 3 に従って、実行可能プロセスキュー 1 1 2 にキューイングされている各ワークロードクラスのプロセスを実行する。具体的には、実行制御手段 1 2 2 は、インターバルを開始すると

50

、インターバルの開始後のCPU使用時間が次のワークロードクラス毎の最大CPU使用時間135以下であるワークロードクラスに属するプロセスにCPUリソースを割り当てて実行する。また実行制御手段122は、インターバルの開始後のCPU使用時間が上記最大CPU使用時間以下であるワークロードクラスに属するプロセスが存在しなくなった時点でインターバルを終了する。また実行制御手段122は、インターバル終了時、インターバルの開始から終了までに上記最大CPU使用時間に達したことが原因で行われたCPUリソース割り当ての打ち切り回数とインターバルの開始から終了までの時間である総割り当て時間とを、前回のCPUリソース割り当て打ち切り回数138と前回の総割り当て時間137とに設定する。

#### 【0042】

ステップS3の調整では、調整手段123は、ディスパッチング情報113の前回の総割り当て時間137および前回のCPUリソース割り当て打ち切り回数138に基づいて、前述した次のインターバル時間見直し処理(1)および次のインターバル時間見直し処理(2)によって次のインターバル時間134を決定し、また決定後の次のインターバル時間に基づいて次のワークロードクラス毎の最大CPU使用時間135を決定する。

#### 【0043】

次に本実施形態に係るCPUリソース管理装置100の効果を説明する。

#### 【0044】

例えば、a、b、cの3つのワークロードクラスがあり、そのCPUリソース配分比を4:3:3とし、インターバル時間およびベースインターバル時間を100msとすると、そのインターバルでは、図9の上段に示すように、ワークロードクラスa、b、cに属する各プロセスに、それぞれ最大40ms、30ms、30msのCPU使用時間が割り当てられることになる。しかし、ワークロードクラスa、bが低負荷、ワークロードクラスcが高負荷な場合、例えば図9の中段に示すように、ワークロードクラスa、bに割り当てられた最大CPU使用時間に余剰が生じる一方、ワークロードクラスcでは最大CPU使用時間に達したことが原因でCPUリソース割り当ての打ち切りが生じる。

#### 【0045】

図9の例では、低負荷なワークロードクラスa、bは、それぞれ30ms、20msしか使用せず、それぞれ最大CPU使用時間のうちの10msは未使用である。一方、高負荷なワークロードクラスでは、最大CPU使用時間の30msを全て使った後、CPUリソースの割り当てを打ち切られている。その結果、総割り当て時間は、 $30 + 20 + 30 = 80$ msとなっている。

#### 【0046】

このようなインターバルの実行後に行われる調整手段123による次のインターバル時間見直し処理(1)では、前回の総割り当て時間である80msがベースインターバル時間である100msより小さく、且つ、CPUリソースの割り当てが打ち切られたワークロードクラスcが存在するため、図4の判定ルールから次のインターバル時間を前回のインターバル時間から変更すると判断する。そして、調整手段123は、図5の計算式1の右辺の前回のインターバル時間に100ms、ベースインターバル時間に100ms、前回の総割り当て時間に80msを代入することにより、次のインターバル時間として125msを計算する。次に調整手段123は、次のインターバル時間見直し処理(2)における図6の条件式が成立するか否かを判断するが、ここでは成立しなかったと仮定すると、次のインターバル時間は最終的に125msに決定される。

#### 【0047】

次に調整手段123は、次のインターバル時間である125msを配分比に従って配分することにより、次のワークロードクラス毎の最大CPU使用時間として、図9に示すように、ワークロードクラスa、b、cにそれぞれ50ms、38ms、38msを割り当てる。そして、実行制御手段122は、次のインターバル時間である125ms、次のワークロードクラスa、b、cの最大CPU使用時間である50ms、38ms、38msの条件の下で、次のインターバルの実行を開始する。

10

20

30

40

50

## 【 0 0 4 8 】

ワークロードクラス a、b、c の負荷状況に変化がなければ、次のインターバルにおいて、ワークロードクラス a、b は最大 CPU 使用時間の 50 ms、38 ms の全てを使わず、一部は未使用になる。一方、ワークロードクラス c は、最大 CPU 使用時間の 38 ms を全て使用することになる。その結果、次のインターバルにおいて予想される実際に使用する CPU 使用時間は、図 9 に示すように、ワークロードクラス a は 37 ms ( $30 \times 125 / 100$ )、ワークロードクラス b は 25 ms ( $20 \times 125 / 100$ )、ワークロードクラス c は最大 CPU 使用時間の 38 ms となり、その合計 (即ち、総割り当て時間) は 100 ms となる。換言すれば、調整手段 123 は、前回のインターバルの総割り当て時間をもとに、次のインターバルの総割り当て時間がベースインターバル時間となるように、インターバル時間を長く設定し、これをもとに、各ワークロードクラスの最大 CPU 使用時間を決定していることになる。

10

## 【 0 0 4 9 】

前回および今回のインターバルにおけるプロセス切替回数は共に 3 回であるため、前回と今回のインターバルを合わせると、 $80 \text{ ms} + 100 \text{ ms} = 180 \text{ ms}$  の期間で合計 6 回のプロセス切替が行われることになり、単位時間当たりのプロセス切替回数は、 $6 / 180 = 0.0333$  となる。

## 【 0 0 5 0 】

一方、次のインターバル時間の調整を行わず、インターバル時間を 100 ms に固定すると、図 9 の下段に破線で示すように、次のインターバルにおける総割り当て時間は前回のインターバルと同じく 80 ms となる。そのため、前回と今回のインターバルを合わせると、 $80 \text{ ms} + 80 \text{ ms} = 160 \text{ ms}$  の期間で合計 6 回のプロセス切替が行われることになり、単位時間当たりのプロセス切替回数は、 $6 / 160 = 0.0375 (> 0.0333)$  となる。

20

## 【 0 0 5 1 】

上記のように各ワークロードクラスの CPU 使用状況に応じて、動的に次のインターバル時間を調整することにより、単位時間当たりのプロセス切替回数を低減することができる。その結果、プロセス切替のオーバーヘッドが削減され、システム全体のスループットが向上する。

## 【 0 0 5 2 】

しかし、次のインターバル時間がベースインターバル時間より長くなると、各ワークロードクラスが CPU リソースを割り当てられるまでの待ち時間が長くなるため、応答時間が伸びてしまうという副作用が生じる。そのため、調整手段 123 は、図 4 の判定ルールおよび図 5 の計算式 1 に従い、前回の総割り当て時間がベースインターバルより長い場合には、次のインターバル時間を前回のインターバル時間よりも短い時間に変更することにより、上記副作用を抑えている。

30

## 【 0 0 5 3 】

また、ワークロードクラス数が増大するほどにインターバル内でのプロセス切替回数が増大するため、プロセス切替によるオーバーヘッドが増大する。そのため、調整手段 123 は、図 6 の条件式に基づいて、プロセス切替によるオーバーヘッドが閾値以上となる場合、次のインターバル時間を更に長くすることにより、スループットの低下を防止している。

40

## 【 0 0 5 4 】

以上の説明では、ディスパッチング情報 113 は、前回のインターバル時間 136 および前回の総割り当て時間 137 を記憶するように構成されていた。しかし、ディスパッチング情報 113 は、直前の複数回分のインターバル時間および総割り当て時間の履歴を記憶するように構成されていてもよい。また調整手段 123 は、計算式 1 の右辺における前回のインターバル時間および前回の総割り当て時間として、ディスパッチング情報 113 に記憶された直前の複数回分のインターバル時間および総割り当て時間の平均値を使用するようにしてもよい。

## 【 0 0 5 5 】

50

[ 第 2 の実施形態 ]

図 10 は、本発明の第 2 の実施形態に係る CPU リソース管理装置 200 のブロック図である。図 10 を参照すると、CPU リソース管理装置 200 は、記憶装置 210 と、この記憶装置 210 に接続された複数の CPU 220 - 0 ~ 220 - m とから構成されている。

【 0056 】

記憶装置 210 は、複数の CPU 220 - 0 ~ 220 - m で共用されるハードディスクやメモリなどから構成され、CPU 220 - 0 ~ 220 - m で実行される各種処理に必要な処理情報およびプログラム 214 - 0 ~ 214 - 1 を記憶するように構成されている。プログラム 214 - 0 ~ 214 - 1 は、CPU 220 - 0 ~ 220 - m に読み込まれて実行されることにより各種処理部を実現するプログラムであり、図示しない通信インターフェース部などのデータ入出力機能を介して外部装置（図示せず）や記憶媒体（図示せず）から予め読み込まれて記憶装置 210 に保存される。記憶装置 210 に記憶される処理情報には、ワークロード定義 211、実行可能プロセスキュー 212、および、CPU 220 - 1 ~ 220 - m に 1 対 1 に対応する m 個のディスパッチング情報 213 - 1 ~ 213 - m がある。これらは、図 1 を参照して説明したワークロード定義 111、実行可能プロセスキュー 112、および、ディスパッチング情報 113 と同様に構成されている。

10

【 0057 】

CPU 220 - 0 は、演算処理装置とその周辺回路とから構成され、記憶装置 210 からプログラム 214 - 0 を読み込んで実行することにより、ハードウェアとプログラム 214 - 0 とを協働させて各種処理部を実現するように構成されている。CPU 220 - 0 で実現される処理部には、初期化手段 221 と調整手段 223 とがある。これらは、図 1 を参照して説明した初期化手段 121、調整手段 123 と同様に構成されている。即ち、初期化手段 221 は、ワークロード定義 211 およびディスパッチング情報 213 - 1 ~ 213 - m を初期化するように構成されている。また調整手段 223 は、ディスパッチング情報 213 - 1 ~ 213 - m に記憶された前回のインターバルにおける総割り当て時間および CPU リソース割り当て打ち切り回数に基づいて次回のインターバル時間およびワークロードクラス毎の最大 CPU 使用時間を決定してディスパッチング情報 213 - 1 ~ 213 - m を更新するように構成されている。

20

【 0058 】

CPU 220 - 1 ~ 220 - m は、演算処理装置とその周辺回路とから構成され、記憶装置 210 ~ プログラム 214 - 1 を読み込んで実行することにより、ハードウェアとプログラム 214 - 1 とを協働させて各種処理部を実現するように構成されている。CPU 220 - 1 ~ 220 - m で実現される処理部には、実行制御手段 222 - 1 ~ 222 - m がある。これらは、図 1 を参照して説明した実行制御手段 122 と同様に構成されている。即ち、実行制御手段 222 - 1 ~ 222 - m は、対応するディスパッチング情報 213 - 1 ~ 213 - m に従って、実行可能プロセスキュー 112 にキューイングされている各ワークロードクラスのプロセスを実行するように構成されている。

30

【 0059 】

次に、図 8 を借用して、本実施形態に係る CPU リソース管理装置 200 の動作について説明する。

40

【 0060 】

CPU リソース管理装置 200 は、起動すると、まず初期化手段 221 が動作してワークロード定義 211 および各ディスパッチング情報 213 - 11 ~ 213 - m の初期化を行う（ステップ S1）。次に、各実行制御手段 222 - 1 ~ 222 - m が、対応するディスパッチング情報 213 - 1 ~ 213 - m に従って、インターバルを開始し、実行可能プロセスキュー 212 に格納されたプロセスを CPU 220 - 1 ~ 220 - m 上で実行する制御を行う（ステップ S2）。また何れかの実行制御手段 222 - i（i は 1 から m の何れか）によってインターバルが終了する毎に、調整手段 223 は、可及的速やかに、その実行制御手段 222 - i に対応するディスパッチング情報 213 - i に記憶された前回の

50

インターバルの状況（前回の総割り当て時間、前回のCPUリソース割り当て打ち切り回数）に基づいて次回のインターバル時間およびワークロードクラス毎の最大CPU使用時間を調整する（ステップS3）。そして再び、実行制御手段222-iは、可及的速やかに、調整後のインターバル時間およびワークロードクラス毎のCPU使用時間が記憶されたディスパッチング情報213-iに基づくインターバルを開始してプロセスの実行制御を行う（ステップS2）。このステップS2、S3のループは、システムの運用が停止されるまで継続して実行される。

【0061】

このように本実施形態に係るCPUリソース管理装置200によれば、第1の実施形態に係るCPUリソース管理装置100と同様の理由により、プロセス切替のオーバーヘッドが削減され、システム全体のスループットが向上する。

10

【0062】

また本実施形態に係るCPUリソース管理装置200によれば、複数のCPU22-1～220-mが互いに並行して複数の実行制御手段222-1～222-mを実行するため、単位時間当たりのプロセス実行回数が増大し、システム全体のスループットが向上する。

【0063】

以上の説明では、CPU220-0は実行制御手段を備えていないが、CPU220-1～220-m上の実行制御手段と同様な実行制御手段をCPU220-0に備えるようにしてもよい。また以上の説明では、CPU220-1～220-mは調整手段を備えていないが、CPU220-0上の調整手段と同様な調整手段を各CPU220-1～220-mに備え、CPU220-i（iは1からmの何れか）は、そのCPU220-i上の実行制御手段の処理が終了したときに、そのCPU220-i上の調整手段が調整処理を行い、その後そのCPU220-i上の実行制御手段が次のインターバルを開始するように構成してもよい。

20

【0064】

[第3の実施形態]

図11は、本発明の第3の実施形態に係るCPUリソース管理装置300のブロック図である。図11を参照すると、CPUリソース管理装置300は、記憶装置310と、この記憶装置310に接続されたプロセッサ320とから構成されている。

30

【0065】

記憶装置310は、ワークロード定義311と実行可能プロセスキュー312とディスパッチング情報313とを記憶するように構成されている。ワークロード定義311は、ワークロードクラス毎のCPU使用時間の配分比を表す情報を記憶する。実行可能プロセスキュー312は、何れかのワークロードクラスに属する複数のプロセスを記憶する。ディスパッチング情報313は、ワークロードクラス毎の最大CPU使用時間、ワークロードクラス毎の最大CPU使用時間の和であるインターバル時間、インターバルの開始から終了までに最大CPU使用時間に達したことが原因で行われたCPUリソース割り当ての打ち切り回数、インターバルの開始から終了までの時間である総割り当て時間、ベースインターバル時間を記憶する。

40

【0066】

プロセッサ320は、1以上のCPUを含んで構成されている。またプロセッサ320は、実行制御手段321と調整手段322とを備えている。

【0067】

実行制御手段321は、ディスパッチング情報313に従ってインターバルを開始すると、インターバルの開始後のCPU使用時間が最大CPU使用時間以下であるワークロードクラスに属するプロセスにCPUリソースを割り当てて実行し、インターバルの開始後のCPU使用時間が最大CPU使用時間以下であるワークロードクラスに属するプロセスが存在しなくなった時点でインターバルを終了するように構成されている。また実行制御手段321は、インターバルの開始から終了までに最大CPU使用時間に達したことが原

50

因で行われたCPUリソース割り当ての打ち切り回数と、インターバルの開始から終了までの時間である総割り当て時間とをディスパッチング情報313に記憶するように構成されている。

【0068】

調整手段322は、ディスパッチング情報313に記憶された総割り当て時間がベースインターバル時間より短く、且つ、ディスパッチング情報313に記憶されたCPUリソース割り当ての打ち切り回数が1以上の場合、ディスパッチング情報313に記憶されたインターバル時間をより長い時間に変更するように構成されている。また調整手段322は、ディスパッチング情報313に記憶されたワークロードクラス毎の最大CPU使用時間を、上記より長い時間に変更した後のインターバル時間をワークロードクラス毎のCPU使用時間の配分比に従って配分して計算したワークロード毎の最大CPU使用時間に変更するように構成されている。また調整手段は、変更後のディスパッチング情報313に従って実行制御手段321に次のインターバルを開始させるように構成されている。

10

【0069】

このように構成されたCPUリソース管理装置300は以下のように動作する。即ち、実行制御手段321は、ディスパッチング情報313に従ってインターバルを開始すると、インターバルの開始後のCPU使用時間が最大CPU使用時間以下であるワークロードクラスに属するプロセスにCPUリソースを割り当てて実行し、インターバルの開始後のCPU使用時間が最大CPU使用時間以下であるワークロードクラスに属するプロセスが存在しなくなった時点でインターバルを終了する。また実行制御手段321は、インターバルの開始から終了までに最大CPU使用時間に達したことが原因で行われたCPUリソース割り当ての打ち切り回数と、インターバルの開始から終了までの時間である総割り当て時間とをディスパッチング情報313に記憶する。

20

【0070】

次に調整手段322は、ディスパッチング情報313に記憶された総割り当て時間がベースインターバル時間より短く、且つ、ディスパッチング情報313に記憶されたCPUリソース割り当ての打ち切り回数が1以上の場合、ディスパッチング情報313に記憶されたインターバル時間をより長い時間に変更する。また調整手段322は、ディスパッチング情報313に記憶されたワークロードクラス毎の最大CPU使用時間を、上記より長い時間に変更した後のインターバル時間をワークロードクラス毎のCPU使用時間の配分比に従って配分して計算したワークロード毎の最大CPU使用時間に変更する。また調整手段は、変更後のディスパッチング情報313に従って実行制御手段321に次のインターバルを開始させる。

30

【0071】

以上のように構成され動作するCPUリソース管理装置300によれば、コンピュータシステムのスループットを改善することができる。その理由は、前回のインターバルが、最大CPU使用時間のCPUリソースを必要としないワークロードクラスが存在したことによりベースインターバルよりも短時間で終了し、かつ、最大CPU使用時間以上のCPUリソースを必要としたワークロードクラスが存在していた場合、次のインターバルを前回のインターバルよりも長い時間に変更し、その変更に合わせてワークロードクラス毎の最大CPU使用時間も変更するため、各ワークロードクラスの負荷が前回と同様であれば、次の総割り当て時間が前回の総割り当て時間よりも長くなり、単位時間当たりのプロセス切替回数が低減し、その結果、プロセス切替のオーバーヘッドが削減されるためである。

40

【0072】

以上、上記各実施形態を参照して本発明を説明したが、本発明は、上述した実施形態に限定されるものではない。本発明の構成や詳細には、本発明の範囲内で当業者が理解しうる様々な変更をすることができる。

【産業上の利用可能性】

【0073】

50

本発明は、予め決められたCPUリソースの利用割合に従って、各ワークロードで使用するCPUリソースの配分を行うCPUリソース管理といった用途に適用できる。

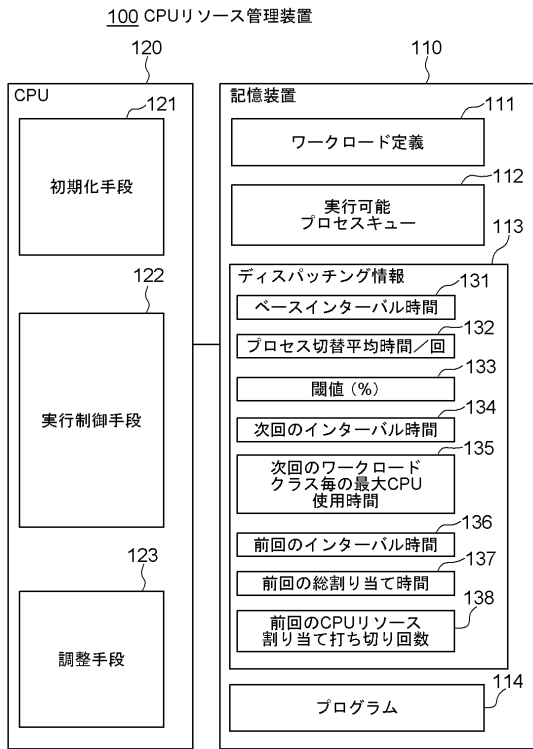
【符号の説明】

【0074】

100 ... CPUリソース管理装置	
110 ... 記憶装置	
111 ... ワークロード定義	
112 ... 実行可能プロセスキュー	
113 ... ディスパッチング情報	
114 ... プログラム	10
120 ... CPU	
121 ... 初期化手段	
122 ... 実行制御手段	
123 ... 調整手段	
131 ... ベースインターバル時間	
132 ... プロセス切替平均時間	
133 ... 閾値	
134 ... 次のインターバル時間	
135 ... 次のワークロードクラス毎の最大CPU使用時間	
136 ... 前回のインターバル時間	20
137 ... 前回の総割り当て時間	
138 ... 前回のCPUリソース割り当て打ち切り回数	
200 ... CPUリソース管理装置	
210 ... 記憶装置	
211 ... ワークロード定義	
212 ... 実行可能プロセスキュー	
213 - 1 ~ 213 - m ... ディスパッチング情報	
214 - 0 ~ 214 - 1 ... プログラム	
220 - 0 ~ 220 - m ... CPU	
221 ... 初期化手段	30
222 - 1 ~ 222 - m ... 実行制御手段	
223 ... 調整手段	
300 ... CPUリソース管理装置	
310 ... 記憶装置	
311 ... ワークロード定義	
312 ... 実行可能プロセスキュー	
313 ... ディスパッチング情報	
320 ... プロセッサ	
321 ... 実行制御手段	
322 ... 調整手段	40

【図面】

【図 1】



【図 2】

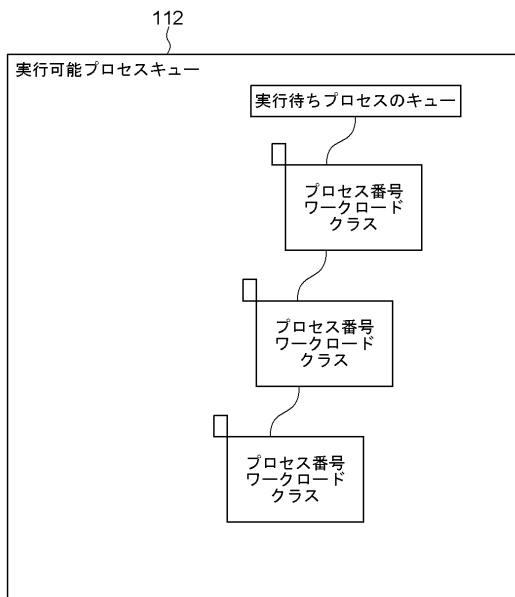
111 ワークロード定義

クラス定義	配分定義
ワークロードクラス a	4
ワークロードクラス b	3
ワークロードクラス c	3

10

20

【図 3】



【図 4】

前回の総割り当て時間： ベースインターバル時間	CPUリソースの割り当てが 打ち切られたワークロードクラス	
	存在する	存在しない
<	変更する	変更しない
=	変更しない	変更しない
>	変更する	変更する

30

40

50

【 図 5 】

【 図 6 】

$$\frac{\text{次回のインターバル時間}}{\text{前回のインターバル時間}} = \frac{\text{前回のインターバル時間}}{\text{前回の総割り当て時間}} \times \frac{\text{ベースインターバル時間}}{\text{前回の総割り当て時間}}$$

... (1)

$$\frac{\text{プロセス切替平均時間}}{\text{ベースインターバル時間}} \times n \text{回} > \text{閾値}$$

... (2)

10

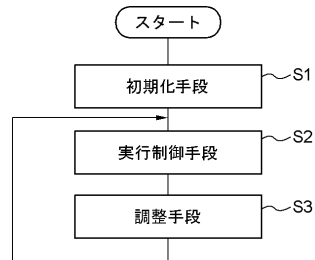
20

【 図 7 】

【 図 8 】

$$\frac{\text{次回のインターバル時間}}{\text{前回のインターバル時間}} = \frac{\text{次回のインターバル時間見直し処理(1)で決定した次回のインターバル時間}}{\text{前回のインターバル時間}} \times \frac{\text{プロセス切替平均時間}}{\text{ベースインターバル時間}} \times n \text{回} \div \text{閾値}$$

... (3)

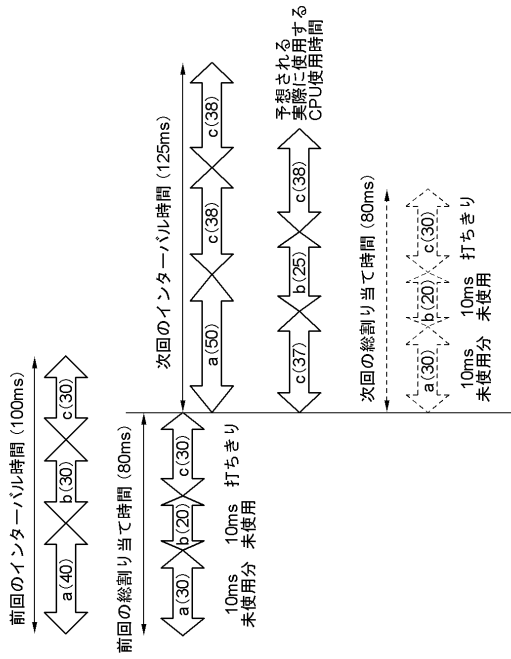


30

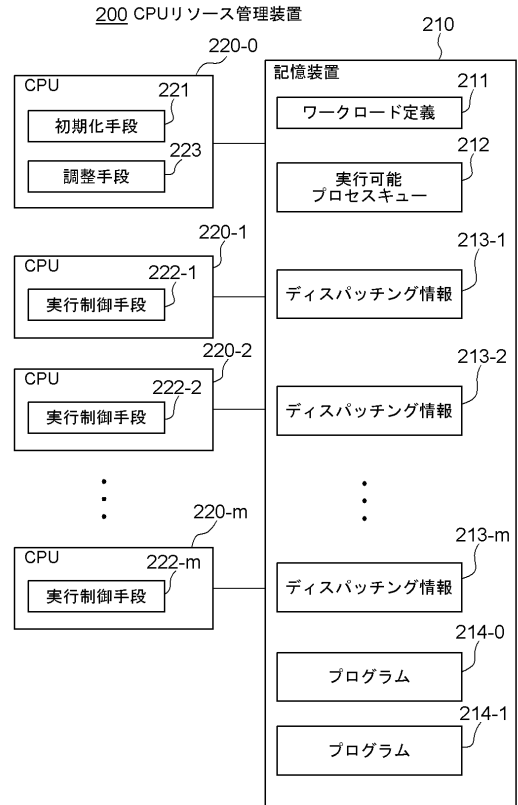
40

50

【図 9】



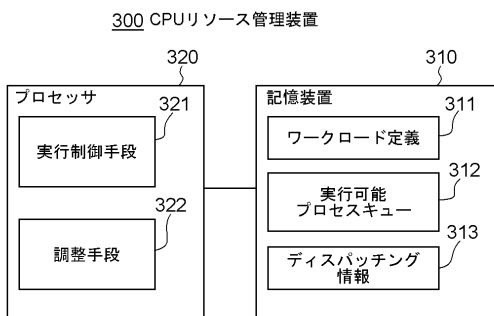
【図 10】



10

20

【図 11】



30

40

50

## フロントページの続き

社内

審査官 坂東 博司

- (56)参考文献 特開平05 - 241860 (JP, A)  
特開昭62 - 011941 (JP, A)  
特開平04 - 336334 (JP, A)  
特開平08 - 147178 (JP, A)  
特開2008 - 225641 (JP, A)
- (58)調査した分野 (Int.Cl., DB名)  
G06F 9/48