

(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2004-62911

(P2004-62911A)

(43) 公開日 平成16年2月26日(2004.2.26)

(51) Int. Cl.<sup>7</sup>

G06F 9/46

F I

G06F 9/46 350

テーマコード(参考)

5B098

審査請求 未請求 請求項の数 10 O L (全 18 頁)

(21) 出願番号 特願2003-278843 (P2003-278843)  
 (22) 出願日 平成15年7月24日(2003.7.24)  
 (31) 優先権主張番号 10/206,594  
 (32) 優先日 平成14年7月26日(2002.7.26)  
 (33) 優先権主張国 米国(US)

(特許庁注: 以下のものは登録商標)

1. イーサネット
2. フロッピー

(71) 出願人 503003854  
 ヒューレット-パッカード デベロップメント カンパニー エル. ピー.  
 アメリカ合衆国 テキサス州 77070  
 ヒューストン 20555 ステイト  
 ハイウェイ 249  
 (74) 代理人 100081721  
 弁理士 岡田 次生  
 (74) 代理人 100105393  
 弁理士 伏見 直哉  
 (74) 代理人 100111969  
 弁理士 平野 ゆかり

最終頁に続く

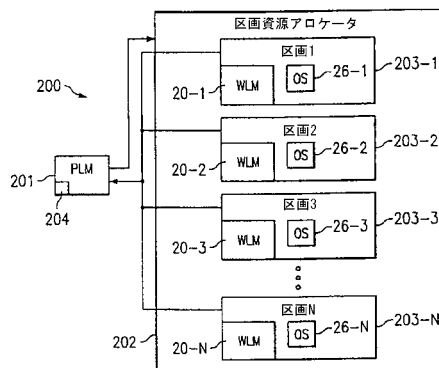
(54) 【発明の名称】 コンピュータ資源の割り当てを管理するシステム

(57) 【要約】

【課題】 コンピュータ資源の、複数区画コンピュータシステムのうちの複数の区画のうち少なくとも1つの区画への割り当てを管理するシステムおよび方法を提供すること。

【解決手段】 複数の作業負荷マネージャを有し、1つの作業負荷マネージャが複数の区画のうち各区画に関連付けられ、各作業負荷マネージャが、コンピュータ資源に関連付けられた区画に割り当てられた少なくとも1つの優先度に基づいて、コンピュータ資源の資源要求値を決定する。本システムは区画負荷マネージャも有し、区画負荷マネージャは、各資源要求値に基づいて各区画それぞれの割り当て値を形成するように動作する。本システムは、割り当て値に基づいて複数の区画にコンピュータ資源を配分する。

【選択図】 図2A



## 【特許請求の範囲】

## 【請求項 1】

複数の区画を有するコンピュータシステムの複数の区画のうちの少なくとも1つへのコンピュータ資源の割り当てを管理するシステムであって、

前記複数の区画のそれぞれに1つが関連付けられた複数の作業負荷マネージャを備え、  
前記コンピュータ資源に関連する区画に割り当てられた少なくとも1つの優先度に基づいて、それぞれの作業負荷マネージャが前記コンピュータ資源の資源要求値を決定し、  
それぞれの資源要求値に基づいて、前記それぞれの区画に割り当て値を形成する区画負荷マネージャを備え、

前記割り当て値に基づいて、前記複数の区画に前記コンピュータ資源を配分するようにした前記システム。 10

## 【請求項 2】

少なくとも1つのパフォーマンスモニタは、前記複数の区画のそれぞれの区画に関連付けられる複数のパフォーマンスモニタをさらに備え、各パフォーマンスモニタには、前記区画のそれぞれの特徴が関連付けられる請求項1記載のシステム。

## 【請求項 3】

少なくとも1つのパフォーマンスモニタによって提供される情報が、前記コンピュータ資源に資源要求値を決定するにあたり前記作業負荷マネージャによって使用される請求項2記載のシステム。

## 【請求項 4】

前記区画負荷マネージャの構成は、

前記区画作業マネージャであって、前記複数の区画のうちの1つの区画に常駐し、前記複数の区画のうちの残りの区画にアクセスできるものと、

前記区画負荷マネージャであって、前記複数の区画の各区画に常駐するものと、

前記区画負荷マネージャであって、前記複数の区画とは別個のモジュールに常駐するものと、  
によりなる群から選択される請求項1記載のシステム。

## 【請求項 5】

前記区画負荷マネージャは、累積的に丸めて非整数割り当て要求値を整数に調整するラウンドを備える請求項1記載のシステム。

## 【請求項 6】

前記区画負荷マネージャは、前記資源要求値の前記優先度に基づいて前記資源要求値を優先度グループへとグループ化し、

優先度グループの各区画が受け取る前記コンピュータ資源の量を等化すること、

それぞれの要求量の前記コンピュータ資源を受け取る区画の数を、優先度グループ内で最大化すること、

優先度グループの前記区画それぞれの前記割り当て値および前記要求量の割合を等化すること、

によりなる群から選択される所定の分配様式に基づいて、割り当て値を形成する請求項1記載のシステム。

## 【請求項 7】

複数の区画を有するコンピュータシステムの複数の区画のうちの少なくとも1つの区画へのコンピュータ資源の割り当てを管理する方法であって、

前記コンピュータ資源に関連する各区画に割り当てられた少なくとも1つの優先度に基づいて、前記複数の区画のうちの各区画に前記コンピュータ資源についての資源要求値を決定することと、

それぞれの資源要求値に基づいて、前記区画それぞれに割り当て値を形成することと、

前記割り当て値に基づいて、前記複数の区画に前記コンピュータ資源を配分することと、  
を含む方法。

## 【請求項 8】

前記割り当て値を形成するステップは、

10

20

30

40

50

前記資源要求値の前記優先度に基づいて、前記資源要求値を優先度グループにグループ化することと、

( a ) 未検査の最高優先度グループ中の前記資源要求値を調べることと、を含む請求項 7 記載の方法。

【請求項 9】

前記割り当て値を形成するステップは、

( b ) 前記コンピュータ資源のうちの要求量を、未検査の最高優先度グループ中の各区画に割り当てることができるかどうかを判断することを含む請求項 8 記載の方法。

【請求項 10】

前記割り当て値を形成するステップは、

( c ) 前記コンピュータ資源のうちの前記要求量を、未検査の最高優先度グループ中の各区画に割り当てることができる場合、前記資源要求値それぞれにおける前記要求量を各割り当て値に等しく割り当てることを含む請求項 9 記載の方法。

10

【発明の詳細な説明】

【技術分野】

【0001】

本願はコンピュータシステムに関し、特に、アプリケーション間でのコンピュータ資源の動的割り当てに関する。

【0002】

( 関連出願の相互参照 )

20

本願は、2000年1月28日付で出願された「DYNAMIC MANAGEMENT OF COMPUTER WORKLOADS THROUGH SERVICE LEVEL OPTIMIZATION」と題する米国特許出願第09/493,753号の一部継続出願であり、2000年4月29日付で出願された「RECONFIGURATION SUPPORT FOR A MULTI PARTITION COMPUTER SYSTEM」と題する、本願と同一の譲受人に譲渡された係属中の米国特許出願第09/562,590号に関連する。

【背景技術】

【0003】

基本的にコンピュータシステムは、限られた資源、特にCPU資源を有する。これら限られた資源を、システム内で動作している様々なアプリケーションに割り当てなければならない。システム資源をアプリケーションに割り当てる従来の割り当て機構は、図1Aに示すプロセス資源マネージャ(PRM)10として知られるシステムである。このシステムは、CPU資源11および他の各種資源を様々なアプリケーション12、13、14に分割するために使用される。PRMは、図1Bに示すPRM構成中の割合として表されるように、資源を全体の何分かの一に分割する。次いで、その各部分または各一部を、アプリケーションを含むプロセス群に割り当てる。そして、各アプリケーションが利用可能な資源のいくらかの部分を受け取る。

30

【0004】

PRMは静的な機構である。すなわち割り当て構成が管理者によって固定され、管理者によってのみ変更可能である。言い換えれば、管理者が、区画をどこに置くか、すなわち、マシンの何パーセントをアプリケーション12、アプリケーション13、およびアプリケーション14に割り当てるかを指定する。

40

【0005】

米国特許第5,675,739号では、アプリケーションの処理に優先度ベースモデルを使用する機構が開示されている。この場合、優先度の高いアプリケーションは、優先度のより低いアプリケーションより先に待ち行列からサービスを受ける。この機構では、処理パフォーマンスを調整するように優先度を変更することができる。

【特許文献1】米国特許第5,675,739号

【発明の開示】

【発明が解決しようとする課題】

【0006】

50

指定情報は固定なので、様々なアプリケーションのニーズの変化に応じることができない。たとえば、アプリケーションによっては、殆どアイドル状態であっても、時々大量の作業を行うものがある。割り当てが固定された静的機構の下では、このアプリケーションへのCPU資源の割り当てが比較的少なくなる。これは、アイドル時間が長いので割合の資源の割り当てを大きくすることが認められないからである。したがって大量の作業を受けた場合、割り当てが少ないのでそのアプリケーションのパフォーマンスは下がる。そして、トランザクションの処理により長い時間がかかる。別の例では、トランザクションがより長期の時間にわたって大量の資源が必要になるが、アイドル時間もある。割り当てが固定された静的機構の下では、このアプリケーションにはCPU資源がより多く割り当てられる。この場合、このアプリケーションがアイドル状態のとき、このアプリケーションに、使用されないのに大量の資源が割り当てられる。従って他のアプリケーションではその資源を利用できないので、他のアプリケーションのパフォーマンスが下がる。このため、その他のトランザクションの処理により長い時間がかかることになる。このように、この機構では、様々なアプリケーションで要件が変化する場合に対応できない。

10

**【0007】**

また管理者が資源を分割するという別の問題がある。管理者は、実際のマシンについて様々なアプリケーションの資源および要件を考えなければならない。問題なのは人間が通常見るよりも、資源およびアプリケーションが低いレベルで動作するからである。さらに、管理者は、どこに区画(partition、パーティション)を設定するかを判断するために、アプリケーションの特徴およびその資源要件について深い知識を有する必要がある。知識のない部分は、当て推量で通常ごまかされる。たとえば、管理者が、アプリケーション13に20%のCPU資源を設定するように選んだとする。システムのユーザが不満を訴えた場合、管理者は後にその値を変更することができる。

20

**【0008】**

他の機構が、IBMによる米国特許第5,675,739号に教示されている。IBMの機構は、アプリケーションの処理に優先度ベースモデルを使用する。言い換えれば、優先度の高いアプリケーションは、優先度のより低いアプリケーションより先に待ち行列からサービスを受ける。この機構は、処理パフォーマンスを調整するように優先度を変更することができる。

**【0009】**

かかる従来技術による機構は、複数区画システムで用いる場合も効率が悪い。大規模コンピュータシステム、たとえば複数のプロセッサ、複数のI/O資源、複数の記憶資源等を有するコンピュータシステムは、区画または保護領域に分けることができる。これら区画は、ハードウェアを区切るものであり、区切ることで資源を別個の機能ブロックに配置する。あるブロック中の資源は、別のブロック中の資源に直接アクセスすることができない。これにより、あるアプリケーションがすべてのシステム資源を使用できず、故障および誤りが発生する。しかし、区画は、一度画定されると静的な性質であり、オペレータの介入なく容易に変更できない。したがって、資源をある区画から別の区画に移し、作業負荷平衡を満足させることは容易ではない。

30

**【0010】**

本発明の目的は、コンピュータ資源の、複数区画コンピュータシステムのうちの複数の区画のうち少なくとも1つの区画への割り当てを管理するシステムおよび方法を提供することである。

40

**【課題を解決するための手段】****【0011】**

本システムは複数の作業負荷マネージャを有し、1つの作業負荷マネージャが複数の区画のうち各区画に関連付けられ、各作業負荷マネージャが、コンピュータ資源に関連付けられた区画に割り当てられた少なくとも1つの優先度に基づいて、コンピュータ資源の資源要求値を決定する。本システムは区画負荷マネージャも有し、区画負荷マネージャは、各資源要求値に基づいて各区画それぞれの割り当て値を形成するように動作する。本シ

50

システムは、割り当て値に基づいて複数の区画にコンピュータ資源を配分する。

【発明を実施するための最良の形態】

【0012】

本発明は、コンピュータシステムにおける作業負荷特徴の変化に動的に応じることを目的としている。コンピュータシステムは、単一の小型コンピュータ、たとえばパーソナルコンピュータ、単一の大型コンピュータ（たとえば、企業サーバ）、あるいはより大型のコンピュータおよび/または小型コンピュータのネットワークを含むことができる。コンピュータ、特に大型コンピュータ、またはネットワークは保護領域または区画に分割することができる。各区画は、それぞれのオペレーティングシステムを走らせることができる。いずれの場合でも、本発明の機構は、管理者がコンピュータシステム資源および要件ではなくパフォーマンスを目標にするのが好ましい。したがって、管理者は、優先度の異なる様々なパフォーマンス目標を定めるのが好ましく、本発明の機構が、資源の必要な調整をいずれも行うのが好ましい。最終的には、区画に関係なく設定することができるのが好ましい。たとえば、コンピュータシステムのデータベース部分の目標は、検索トランザクションにかかる時間が10ミリ秒を越えないものとするることができる。そして、本発明の機構が、この目標を達成するように資源を操作する。複数区画コンピュータシステムの場合、資源は区画内で操作される。たとえば、プロセッサ時間をアプリケーションに割り当てるか、区画間で操作、たとえばある区画から他の区画へのプロセッサの再割り当てを行う（効率的な区画のサイズ変更）か、またはこれら双方の組み合わせとすることができる。なお、資源は、1台の物理的なコンピュータにあってもよく、あるいは別の物理的なコンピュータにあるアプリケーションまたは区画に割り当てることができる。

【0013】

本発明の機構は、資源要求情報をシステム区画から受け取る区画負荷マネージャ（PLM）を備えるのが好ましい。PLMは、資源要求情報を調べ、要求情報を利用可能な資源と比較するのが好ましい。比較に基づいて、PLMは、特定区画の資源を増加、減少、または変更しないままにできる。区画のパフォーマンスが遅い場合、たとえばトランザクションにかかる時間が目標よりも長い場合、その区画はPLMからの資源割り当ての増大を要求することができる。区画が目標を上回っている場合、その区画は過度の資源を有している旨をPLMに通知することができ、PLMはその区画の割り当てを下げ、下げた分を別の区画（1つまたは複数）に割り当てることができる。

【0014】

各区画は、PLMと同様に動作しつつも特定の区画内で動作する、作業負荷マネージャ（WLM）を含むのが好ましい。WLMについては、2000年1月28日付で出願された「DYNAMIC MANAGEMENT OF COMPUTER WORKLOADS THROUGH SERVICE LEVEL OPTIMIZATION」と題する米国特許出願第09/493,753号により十分に説明されている。各WLMはまた、ユーザすなわち管理者から目標情報および優先度情報を受け取る。なお、かかる目標および優先度情報はすべての区画について同じであっても、または各区画または区画群毎に固有であってもよい。WLMはまた、区画内のアプリケーションおよび装置のパフォーマンスを監視するプロセスであるパフォーマンスモニタから、パフォーマンス情報を受け取る。WLMは、パフォーマンスモニタからの情報を調べ、この情報を目標と比較する。比較に基づいて、WLMは、アプリケーションの割り当てを増加、減少、または変更しないままにできる。アプリケーションのパフォーマンスが遅い場合、たとえば、トランザクションにかかる時間が目標よりも長い場合、WLMは割り当てを増大する。アプリケーションが目標を上回っている場合、WLMは、そのアプリケーションへの割り当てを下げ、下げた分を別のアプリケーションに割り当てる。

【0015】

WLMはまた、PLMと対話する。各WLMは最初に、また定期的に、各自の資源必要性を決定し、資源要求情報をPLMに送る。PLMは、かかる要求を受け取った後、システム資源を区画に割り当てる。各WLMは、区画資源についての情報を受け取った後、割り当てられた資源を区画内のアプリケーションに割り当てる。

10

20

30

40

50

## 【0016】

複数区画システムでは、PLMは一区画に常駐して、その他の区画にアクセスすることができる。あるいは、PLMは、すべての区画を管理するサービスモジュールに常駐することができる。あるいは、PLMは各区画に常駐し、協働して資源を各区画に割り当てることができる。

## 【0017】

区画アービタまたは区画資源アロケータは、区画の優先度および資源要求に基づいて資源を様々な区画に割り当てる。この資源の移動は区画サイズ変更と呼ばれる。区画は、それぞれのWLMを通して、要求される各資源の量が表示された優先度付きアプリケーション目標リストを保持することが好ましい。優先度の等しいアプリケーション目標は同等に扱われる。(アプリケーションが2つ以上の目標を有しうることに留意する)。優先度のより低いアプリケーション目標より前に、優先度のより高いアプリケーションの要求が満たされる。割り当てられていない資源は、予備にすることができ、またはデフォルト区画に割り当てることができる。なお、デフォルト区画のアプリケーションでは、それぞれの目標を常に越えるので、この条件を要因にしては、資源の再割り当てまたは区画のサイズ変更を行わないという規則を要求できる。

10

## 【0018】

なお、区画資源割り当ては、もはや固定された構成ではない。区画の必要性が変化するにつれ、本発明は、資源の可用性および優先度に基づいて区画の割り当てを自動的に調整する。したがって、本発明は動的である。また、本発明は、規定された目標の達成に正しい資源割り当てを判断し、本発明を使用するコンピュータシステムは、記載のパフォーマンス目標を達成する特定の区画割り当て値に収束するので、管理者は初期割り当てを推測するする必要はもはやない。さらに、優先度を様々な目標に割り当てることができる。したがって、システム資源に基づいて様々な目標を満たすことができる。たとえば、大量の資源の場合にはすべての目標を満たすことができるが、資源の量が少なくなるにつれて、より優先度の低い目標よりも先に、より優先度の高い目標が満たされることになる。さらに、システムへの変更は、PLMが資源要求を受け取るとすぐに行うことができ、システム管理者による作業は必要ない。また、複数区画システムでは、管理者が定義し優先順位付けできる目標は、すべての区画、および区画で動作している様々なオペレーティングシステムインスタンスにわたって適用されるものであり、単一の区画内のみ適用されるものではない。

20

30

## 【0019】

図2Aは、複数の区画203-1、203-2、203-3、・・・、203-Nを有する複数の区画における本発明の各種構成要素を示す。各区画は、1つまたは複数のプロセッサと、他のシステム資源、たとえば、記憶装置、I/O装置等を有することができる。各区画は、それぞれのオペレーティングシステム26-1、・・・、26-Nを走らせており、区画間の分離および生存性を提供するのが好ましい。なお、様々な区画が様々な量の資源、たとえば様々な数のプロセッサを有しうる。また、複数区画は1つまたは複数の物理的なコンピュータにあってもよい。区画は仮想的であってもよい。

## 【0020】

また、初期状態では、システムが区画に等分割した資源を有しうる。あるいは、システムの初期状態は、各区画に最小の資源のみを提供し、余分の資源を予備にとっておく、たとえば、割り当てないでおく、あるいは1つまたは複数の区画にすべて配置できる。PLMおよびWLMの動作により、各区画のアプリケーションに定義された目標および優先度の処理に最も効率的な様式で、システム資源を素早く割り当てる。

40

## 【0021】

コンピュータシステムの資源は、PLM201によって管理される。PLM201は、様々な区画から資源要求を受け取る。要求は、複数の優先度および複数の資源タイプを含むことができる。たとえば要求は、次のように記述することができる。区画が優先度高のアプリケーションをすべて処理するために2つのプロセッサおよび1つの記憶装置を必要

50

とする。優先度高および中のアプリケーションをすべて処理するために4つのプロセッサおよび2つの記憶装置を必要とする。優先度高、中、低のアプリケーションをすべて処理するために7つのプロセッサおよび5つの記憶装置を必要とする。要求は、WLM20-1、・・・、20-Nから発せられる。WLMは、好ましくは、それぞれの目標の達成に必要な資源を合計した後に要求を生成する。PLMは、1つまたは複数の要求を受け取った後、好ましくは、システム資源を検討し、存在する資源、現在の要求、および要求の優先度に基づいて、再割り当てが必要かどうかを判断する。したがって、特定の区画の資源要件が変化した場合、PLMは現在の資源と同様に、特定の区画の新しい要件と、他の区画の既存の要件とを調べ、再割り当てが必要かどうかを判断する。PLMは、システム資源変更後、たとえば、プロセッサが故障した、またはさらなるメモリが追加された等の後に再割り当てを開始することもできる。

10

#### 【0022】

PLMは、好ましくは、資源要求の優先度を調べることによって再割り当てが必要かどうかを判断する。高レベル要求が変更されると、通常、再割り当てが行われる。たとえば、すべての装置資源が区画の高優先度動作の処理に消費される場合、低優先度要求の変更は無視される。一方、たとえば必要な資源がより少ないなど高優先度の要求が変化する場合、資源の再割り当てが行われる。たとえば供給過剰な区画からの余分な資源が、それぞれのアプリケーションの目標および優先度に基づいて他の区画に再割り当てされる。次いで、PLMは、様々な区画のアプリケーションの目標および優先度に基づいて、改訂後資源分配を計算する。次いで、改訂後分配が区画資源アロケータ202に送られる。アロケータ202は、好ましくは、区画をサイズ変更するように動作する。区画のサイズ変更とは、PLM201によって提供される命令に基づいて、資源を1つまたは複数の区画から1つまたは複数の区画に移動することである。かかるアロケータおよび区画サイズ変更の例は、2000年4月29日付けで出願された「RECONFIGURATION SUPPORT FOR A MULTIPARTITION COMPUTER SYSTEM」と題する米国特許出願第09/562,590号に記載されている。

20

#### 【0023】

なおサイズ変更することで、システムが相当のオーバヘッドを起こす場合がある。この場合、ある区画から別の区画に資源を移動すると、利用可能な計算時間が短くなる。このため、PLMによる判断には、PLMが再割り当てを開始する前に、閾値に到達しなければならないことを含めることができる。閾値は、複数の構成要素、たとえば時間、容量を下回る/上回る割合等を含むことができる。たとえば、容量を下回る/上回る割合が小さい場合、再割り当てを行うまでより長い時間にわたって容量を下回る/上回るが、容量を下回る/上回る割合が大きい場合、即座に再割り当てを行う。その結果資源の必要性が小さく一時的に変化しても、システムの再割り当ては行わないようになる。

30

#### 【0024】

図2Bは、パフォーマンス目標および優先度21を有する、本発明のシステムの区画の各種構成要素を示す。目標21は、好ましくは構成ファイルを含む。構成ファイルは、ユーザすなわちシステム管理者によって定義され、関心のあるアプリケーション特徴、測定中のアプリケーション特徴、特徴に関してアプリケーションのパフォーマンスの所望レベル、この目標の達成優先度に関してユーザの好みを記述する。ユーザは、特定の目標を達成すべき期間も指定することができる。たとえば、第1のアプリケーションは第1のデータベースであることができる。この場合ユーザは構成ファイルに、特定タイプのトランザクションが2秒以内で完了し、高い優先度を有するという特徴を指定する。アプリケーションは、同じ特徴について第2の目標を有することもできる。たとえば、同じタイプのトランザクションが30秒以内に完了すべきであり、優先度は低いとする。第2のアプリケーションは、第1のデータベースと同様の目標を有する第2のデータベースであることができる。すなわち、特定タイプのトランザクションを2秒以内で完了し、第1のデータベースと同じ優先度を有する。したがって、資源は、ように2つのアプリケーションに割り当てられ、その結果優先度の高い目標を満たす。すべての余剰資源は、第1のアプリケー

40

50

ションに与えられ、その結果優先度のより低い「広がった」目標を満たす。

【0025】

WLM20は、好ましくは、監視中の各アプリケーション12、13、14の特定の特徴(1つまたは複数)の状態を記述するパフォーマンス情報を受け取る。WLM20は、区画208内に含まれるプロセッサ11および他の装置25(たとえば、I/O、記憶装置等)の状態および/または他の特徴を記述するパフォーマンス情報も受け取る。

【0026】

パフォーマンス情報は、好ましくは、パフォーマンスモニタ23によって供給される。図2Bに示すように、単一のモニタは複数のアプリケーションおよび装置を処理することが可能であるが、本発明の異なる実施形態は複数のモニタを備え、それぞれが1つまたは複数のアプリケーションおよび装置を監視してもよい。パフォーマンスモニタ23は、アプリケーションおよび/または装置についての特定の情報を集める小さなプログラムである。たとえば、アプリケーションがデータベースである場合、パフォーマンスモニタはデータベースのアクセス時間を測定する。別の例として、装置がハードドライブの場合、パフォーマンスモニタはデータ容量を測定することができる。情報は厳密にアプリケーションパフォーマンスである必要はなく、作業負荷(たとえば、CPU使用量)の任意の測定可能な特徴であることができる。この情報は、システムが動作している間継続的に収集されている。作業負荷マネージャは、管理者によって指定されるある間隔で情報をサンプリングする。

10

【0027】

作業負荷マネージャの出力は、モニタによって報告される進行中のパフォーマンスおよびユーザにより与えられる目標から導出され、好ましくは、周期的にPRM10に提供される。WLM20の出力は、各アプリケーションに割り当てられる異なる資源の分け前、すなわち割り当て配分である。たとえば、各分け前はCPU動作秒の1/100におおよそ等しい。したがって、アプリケーションが少なくとも1つの実行可能プロセスを有する場合、1秒以内に、10の割り当てを有するアプリケーションは1/10秒を受け取ることになる。受け取る時間は連続していない場合もあり、むしろ1秒間隔で分散される場合があることに留意する。分け前は、割り当てられている資源、たとえば、ディスク記憶空間の割合またはディスク記憶空間の実際のバイト数に基づく他のパラメータに等しくてもよいことに留意する。

20

30

【0028】

区画は、複数の資源、たとえば複数のCPUおよび/または複数の記憶装置を有することができる。したがって、割り当ては、1つの装置にすべて配置されても、または装置にわたって分散してもよい。たとえば、4プロセッサシステムでプロセッサ割り当てを10パーセントにすると、1つのプロセッサで40%、各プロセッサで10%、2つのプロセッサで20%、または他の割り当てにすることができる。異なる装置間での配分は、PRM10が決定する。PRMは、確実に10パーセントを達成する必要があるので、アプリケーションを様々な装置の間で移動させる。したがって、アプリケーションが有する実行可能スレッドが1つのみで1つのCPU上でしか実行できない場合、PRMが与えようとするのは、CPU1つで(2CPUシステム上で)20%、その結果そ外部でCPUを使用できる総領域の10%になる。マルチスレッドアプリケーションは、2つ以上のCPUに割り当てることができる。割り当てにより、アプリケーションをプログラムされたタスクが実行可能になる。タスクを実行する速度および効率は、割り当てられたCPU時間の量を反映する。割り当てられるCPUが少ないほど、時間期間内に実行するタスクは少なくなる。割り当てられるCPUが多いほど、時間期間内に実行するタスクは多くなる。パフォーマンスモニタは、WLMによってサンプリングされることになるパフォーマンスを測定し、システムのフィードバックを完了する。

40

【0029】

WLM20はまた、好ましくは、PLM201への資源要求を送る。これら要求は、リストの形態を採ることができる。リストは、区画208が異なる優先度で目標を達成する

50



のに必要な資源を記述する。次いで、PLMは、要求に基づいて資源を再割り当てすることを決定することができる。PLMは、様々な要求を格納することができる。要求によりPLMは、要求された資源の変化がわかるので、資源の変化を予測できる。たとえば、時間の経過に伴い、PLMは、特定の区画が常に特定の時間（または特定の事象の後に）、たとえば午後4時により多くの資源を必要とすることを認識し、それによって資源をその特定の区画に、区画が要求を送る前に割り当てることができる。要求を格納することで、再割り当てを引き起こす基準の設定できる。単純なトリガを使用して単一のメッセージを現在の資源割り当てと比較することができる。たとえば、現在の資源割り当てを5%以上増加/低減するよう要求されると、再割り当てを行う。より複雑なトリガを使用して、格納メッセージを参照することもできる。たとえば、特定の区画からの、現在の資源割り当ての2%~5%未満の増加/低減する要求が1時間より長く続く場合、再割り当てを行う。

10

#### 【0030】

PLM201は、図3のフローチャート300に従って動作する。PLMは、WLMから資源要求を受け取る(302)ことによって開始する(301)。次いで、オプションとして、PLMは再割り当てを開始するか否かを判断する(315)。PLMは、資源要求を現在の割り当てと比較することができる。特定の区画が、現在の割り当てと比較して、所定の閾値を越えるより多くの、またはより少ない資源を要求する場合、PLMは再割り当てを開始することができる。また、PLMは、時間の経過に伴って累積された各区画からの複数の要求を比較して、資源の慢性的な余剰/不足があるかどうかを判断する。たとえば、要求資源と現在の資源との差分が10%(余剰あるいは不足)の場合、即座に再割り当てが行われる。差分が9%の場合、2つの連続した要求(または10分間)において差分(9%以上)が発生する場合には再割り当てがなされ、差分が8%の場合(8%以上)、3つの連続した要求(または15分間)において差分が発生する場合には再割り当てが行われると仮定する。PLMは、再割り当てを行うべきであると判断する場合にはボックス316に進み、行うべきではないと判断する場合にはボックス302に戻る。

20

#### 【0031】

ボックス316において、PLMは、好ましくは、すべての区画に値1(以下、装置の最小割り当て、たとえば、1つのCPU、1つのI/O、1つのメモリブロック等を意味する)を割り当てる(301)。余分な資源は、デフォルト区画に割り当てられるか、または割り当てられずに予備としておく。あるいは、PLMは、区画に資源を等分割することができる。

30

#### 【0032】

次いで、ボックス303において、PLMは、好ましくは、区画のアプリケーション最高優先度グループの処理に必要な資源要求を調べる。優先度グループ内の各区画の要求量を満たすことができるかどうかを判断する(304)。満たすことができる場合、PLMは、割り当て情報を区画資源アロケータ202に送ることによって、要求割り当ての割り当てを促す(305)。各アプリケーション優先度レベル、区画毎に1つまたは複数のものとともに、いくつかのメッセージを送ることができることに留意する。あるいは、最後(309)に1つのメッセージを送ることにより、すべての区画への資源の完成した割り当てをレイアウトすることができる。要求量を満たすことができない場合、PLMはステップ310に述べるように、均衡に様々な区画の間を調停するのが好ましい。ステップ305においてアプリケーション優先度グループ中の各区画の要求を満たした後、PLMは、アプリケーション優先度グループがまだあるかどうかを判断する(306)。ある場合、PLMはステップ303に戻り、繰り返す。もうない場合、PLMは割り当てられていない資源が残っているかどうかを判断する(307)。残っていない場合、PLMは終了する(309)。割り当てられた資源の情報が区画資源アロケータに送られ、PLMはこの繰り返しについて終了する。新しい要求を受け取った後、PLMはステップ301において再開する。ステップ307において、資源が利用可能であると判断される場合、PLMは、残っている資源をデフォルト区画に割り当てるか、その資源を割り当てないものと

40

50

指定して予備にとっておく(貯蔵(hoarding))か、または残っている資源を区画のうちの1つまたは複数で等しく分けることができる。なお、貯蔵することで、本発明がより適正に動作できるようになる。余分な資源を割り当てることで、区画がそれぞれの目標を上回ることになり、その結果、再割り当てを回避する規則を使用しない限り、さらなる再割り当てにつながるからである。次いで、PLMは終了する(309)。

ステップ304において、アプリケーション優先度グループ内の各区画の要求量を満たすことができないとPLMが判断する場合、PLMは、異なる区画の間を均衡に調停することが好ましい。たとえば、(1)ターゲット値にまだ使用されたことのない、それまで割り当てられた任意の量のうちの最低の割り当て量、または(2)以前のターゲット値にまだ使用されたことのない、優先度グループの中の一区画の最低要求量、のうちの低いほうの値として、現在のターゲット値を指定する(310)という方法がある。なおこれにより、図3に示すPLMのパフォーマンスフローが簡略化されるので(すなわち、ステップ310、311、312、および313が繰り返される回数を低減することによって)、基準(1)および(2)は、それぞれの要求量に達した区画を含まない。次いで、PLMは、アプリケーション優先度グループ内の各区画のターゲット量を満たすことができるかどうかを判断する。満たすことができない場合、割り当て量を、割り当てが現在のターゲットよりも少ないアプリケーション優先度グループの様々な区画に等分することができる(314)が、すでにターゲットレベルを満たしているか、またはターゲットレベルを超えている区画は除外する。次いで、PLMは終了する(309)。満たすことができる場合、PLMは、各区画の資源割り当て値をターゲットレベルにするに十分な資源を割り

10

20

#### 【0033】

なおボックス314の分配は例に過ぎず、残っている量は、予備にとっておくこともできる。または、割り当てられていない資源を、デフォルト区画に割り当てることも、別の規則に従って1つまたは複数の区画に割り当てることもできる。

#### 【0034】

図4Aは、PLM201の動作の例を示す。図4Aに示すように、4レベルの優先度毎に異なる要件を有する6つの区画がある。なお様々なタイプの資源が存在するが、簡便化のために1つのみの資源タイプを示す。各区画は様々なタイプの資源毎に異なる要件をもつことができる。図示のように、区画1は、優先度2および3のアプリケーションまたはプロセスと同様に優先度1のアプリケーションまたはプロセスの処理に、1つの資源を必要とする。そして優先度4のアプリケーションまたはプロセスの処理に3つの資源を必要とする。その他の区画は、それぞれの要件を図示のように有する。これら資源は単一のプロセッサ、プロセッサグループ、I/O装置、メモリ(たとえば、RAM、ROM等)、記憶装置(光ディスク、ハードドライブ等)、他の装置および/またはシステムへの接続帯域(たとえば、インターネット、イントラネット、LAN、WAN、イーサネット等)

30

40

#### 【0035】

なお要件を表すために用いられる値は、例としてのみ資源の増分値として示されるが、他の値を使用することができる。たとえば、記憶装置(RAM、ROM、ハードドライブ等)の場合、要件はメガバイト、またはハードドライブの個数として示すことができる。プロセッサは、パーセンテージ、シェア、または正規化値として示すことができる。コンピュータシステムによっては、小数値を使用することができ、資源が区画の間で分割されることができる。コンピュータシステムが小数値を処理できない(資源の分割がない)場

50

合、丸め誤差または不均衡が資源の割り当てに生じうる。

【0036】

図4Aは、要求時の図3に示すPLMの割り当て動作も示す。すべての区画のすべての優先度に必要な合計資源量は21個であるが、システムには合計19個の資源が存在することに留意する。したがって、すべての区画がそれぞれの優先度を満足させることができるわけではない。時間期間後、区画は、図4Aに表形式で示す資源要求をPLMに送る。次いで、PLMは、ボックス315において再割り当てが必要であると判断し、資源の均衡な割り当てを開始する。システムにさらなる資源を追加する、たとえば、もう1つのプロセッサの追加によっても再割り当てを行うことになりうることに留意する。同様に、システムから資源が除去される、たとえば、I/O装置が故障しても、再割り当てにつながり得る。

10

【0037】

PLMは動作するに最小限の資源を各区画に提供することによって開始し、図3のボックス316に従って、列401に示すように各区画に1つの資源を割り当てる。たとえば、各区画は、動作するために少なくとも1つのプロセッサ、メモリブロック、および1つのI/O装置を備えなければならない。PLMは、資源情報を区画資源アロケータ202に送るか、または資源情報を区画資源アロケータ202に送る前に、再割り当てが完了するまで待つことができる。

【0038】

次いで、PLMは、各区画がそれぞれの要求資源量を優先度1について受け取ることができるかどうかを判断する(ボックス304)。この場合、残りの資源は13個であるため、要求される資源量を割り当てることができる。列402に示すように、区画3および5はそれぞれ1個の追加資源を受け取る(ボックス305)。その他の区画は、最初の割り当てで満たされている。

20

【0039】

さらなる優先度グループがある(ボックス306)ため、PLMは優先度2について繰り返す。11個の資源が残っているため、PLMはここでも要求量を割り当てることができる。したがって、列403に示すように、区画2および3はさらに2個の資源を受け取り、区画5はさらに1個の資源を受け取る。

【0040】

さらなる優先度グループがあるため、PLMは優先度3について繰り返す。PLMは、6個の資源が残っているため、ここでも要求量を割り当てることができる。したがって、列404に示すように、区画2および5がさらに1個の資源を受け取る。

30

【0041】

さらなる優先度グループがあるため、PLMは優先度4について繰り返す。残りの資源が4個しかないため、PLMは要求量を割り当てることができない。区画は、さらに6個の資源の割り当てを望む。(区画4は合計3個の資源を望み、すでに1個の資源が割り当てられているため、さらに必要な資源は2個のみであることに留意する)。したがって、ボックス304から「いいえ」の経路を辿る。現在のステップについてこれまで割り当てられた量は1および4である一方で、要求される量は1、3、4、および5である。現在のターゲットは、要求している区画の最低値であるとともに、以前割り当てられた量の最低値である1と指定される。各区画は少なくとも1個の資源を有するため、列405に示すように、このサイクルでは追加資源は割り当てられない。区画3および6はそれぞれの要求量に達していることに留意する。追加資源が残っている(ボックス313)ため、新しいターゲット、すなわち3(以前使用されていない最低目標)が指定される。列406に示すように、区画1および4はそれぞれ追加資源を受け取るが、区画2および5は変更されないままである。区画1および4はそれぞれの要求量に達したことに留意する。割り当てられた量は、資源割り当て情報として区画資源アロケータ202に提供される。次いで、アロケータ202は区画の資源を操作する。

40

【0042】

50

図4Bは、図4Aの例と同様の、PLM201の動作の別の例を示す。図4Bに示すように、2レベルの優先度毎に異なる要件を有する5つの区画がある。様々なタイプの資源が存在するが、簡便化のために1つのみの資源タイプを示し、各区画は様々なタイプの資源毎に異なる要件を有しうることに留意する。図示のように、区画1は、優先度1のアプリケーションまたはプロセスの処理に1つの資源を必要とし、優先度2のアプリケーションまたはプロセスの処理に9つの資源を必要とする。その他の区画は、それぞれの要件を図示のように有する。区画5は優先度1については4個の資源を必要とするが、優先度2には3個のみの資源を必要とすることに留意する。このような場合、好ましくは、優先度のより高い要求が満たされる。

【0043】

図4Bは、要求時の図3に示すPLMの割り当て動作も示す。すべての区画のすべての優先度に必要な合計資源量は27個であるが、システムには合計24個の資源が存在することに留意する。したがって、すべての区画がそれぞれの優先度を満足させることができるわけではない。時間期間後、区画は、図4Bに表形式で示す資源要求をPLMに送る。次いで、PLMは、ボックス315において再割り当てが必要であると判断し、資源の均衡な割り当てを開始する。

【0044】

PLMは動作するに最小限の資源を各区画に提供することによって開始し、図3のボックス316に従って、列408に示すように各区画に1つの資源を割り当てる。次いで、PLMは、各区画がそれぞれの要求資源量を優先度1について受け取ることができるかどうかを判断する(ボックス304)。この場合、要求量を割り当てることができる。列409に示すように、区画3および5はそれぞれ3個の追加資源を受け取る(ボックス305)。区画5は要求量に達していることに留意する。その他の区画は、最初の割り当てで満たされている。

【0045】

さらなる優先度グループがある(ボックス306)ため、PLMは優先度2について繰り返す。PLMは、要求量を割り当てることができない。したがって、ボックス304から「いいえ」の経路を辿る。これまで割り当てられた量は1および4であり、要求量は2、3、5、8、および9である。現在のターゲットは、要求量およびそれまで割り当てられた量を含む集合の中で最も低い値である1と指定される。各区画は少なくとも1個の資源を有するため、列410に示すように、このサイクルでは追加資源は割り当てられない。追加資源が残っている(ボックス313)ため、新しいターゲット、すなわち2が指定される。列411に示すように、区画1、2、および4はそれぞれ追加資源を受け取る。区画4は要求量に達したことに留意する。追加資源が残っている(ボックス313)ため、新しいターゲット、すなわち3が指定される。列412に示すように、区画1および2はそれぞれ追加資源を受け取る。追加資源が残っている(ボックス313)ため、新しいターゲット、すなわち4が指定される。列413に示すように、区画1および2はそれぞれ追加資源を受け取る。追加資源が残っている(ボックス313)ため、新しいターゲット、すなわち5が指定される。列414に示すように、区画1、2、および3はそれぞれ追加資源を受け取る。区画3が要求量に達したことに留意する。追加資源が残っている(ボックス313)ため、新しいターゲット、すなわち8が指定される。新しいターゲットに見合うように残りの資源を割り当てることはできない(ボックス311)。したがって、残りの資源はボックス314に従って割り当てられる。たとえば、ボックス314に記述するようにそれぞれの要求割り当てにまだ達していない区画の間で残りの資源を等分することができる。このため、3個の残り資源が区画1および2に分けられ、各区画が1.5個の資源を受け取る。割り当て量は、資源割り当て情報として区画資源アロケータ202に提供される。次いで、アロケータ202は区画の資源を操作する。

【0046】

上述したように、資源ユニット全体の代表ではない資源値が使用され、システムが小数単位を処理することができない、たとえば1つのプロセッサの場合、丸め誤差が生じうる

10

20

30

40

50

。 P L M は、図 5 A に示し、図 5 B および図 5 C の例に例示するようにかかる誤差を処理する。図 5 A は、 P L M 2 0 1 の丸め器部分 2 0 4 の動作を示す。上記例は、要求に整数値を使用したため、割り当て値の結果もまた整数であるが、小数または浮動小数、たとえば割り当て値 1 0 . 1 も使用することができる。また、浮動小数は、ステップ 3 1 4 の結果からも生じうる（たとえば、3 個の資源を 2 個の区画で分けると、各区画毎に 1 . 5 個の資源になる）。システムによっては、整数の割り当て値でしか動作しないものもあるため、資源の小数值を切り上げ、または切り捨てる必要がある。これは、資源全体を割り当てる必要がある、区画サイズ変更の際にプロセッサ、ハードドライブ等の増分資源を割り当てる場合にも当てはまる。丸め器 2 0 4 はまず、 P L M から割り当て値 5 1 を受け取る。割り当て値は、図 3 の動作から得られる値である。次いで、丸め器は、前回割り当てられた値を受け取った各割り当て値に加算することにより、受け取った各割り当て値毎に値を累加する。次いで、丸め器は、各累加和から前回の累加和を差し引くことにより、丸めた割り当て値を形成する。たとえば、図 5 B に示すように、3 個の区画は、割り当て値  $R_1 = 3 . 5$ 、 $R_2 = 3 . 5$ 、および  $R_3 = 3 . 0$  を有する。丸め器は、 $R_1$  と 0 を加算することによって  $S_1$  を形成し（このステップは、 $S_1$  に  $R_1$  の値が割り当てられるように変更しうることに留意）てから丸める。ここで、0 以上かつ厳密に 0 . 5 未満の小数值は 0 に切り捨てられ、0 . 5 以上の小数值は 1 に切り上げられる。同様に、丸め器は、 $R_2 + R_1$  を加算して丸めることによって  $S_2$  を形成し、 $R_3 + R_2 + R_1$  を加算して丸めることによって  $S_3$  を形成する。あらゆる小数值は次の和（丸める前）に累積される、すなわち  $S_1$  は 0 . 5 を有し、 $S_2$  は 1 . 0 を有し、 $S_3$  も 1 . 0 を有する（丸める前）ことに留意する。丸め器は、和から前の和を差し引くことによって、丸められた割り当て値を形成する。具体的には、 $R_1' = S_1$ （すなわち  $S_1 - 0$ ）、 $R_2' = S_2 - S_1$ 、および  $R_3' = S_3 - S_2$  である。最初の値では、累積された小数值が 0 . 5 以上である場合であるため、切り上げが行われることに留意する。次いで、これら丸められた値が区画資源アロケータ 2 0 2 に送られる。

10

20

#### 【 0 0 4 7 】

図 5 C は、4 個の区画が割り当て値  $R_1 = 1 0 . 1$ 、 $R_2 = 2 0 . 2$ 、 $R_3 = 3 0 . 3$ 、および  $R_4 = 3 9 . 4$  を有する丸めの別の例である。丸め器は、 $S_1 = R_1$ （すなわち  $R_1 + 0$ ）および丸めによって  $S_1$  を形成し、 $S_2 = R_2 + R_1$ （すなわち  $R_2 + S_1$ ）および丸めによって  $S_2$  を形成し、 $S_3 = R_3 + R_2 + R_1$ （すなわち  $R_3 + S_2$ ）および丸めによって  $S_3$  を形成し、 $S_4 = R_4 + R_3 + R_2 + R_1$ （すなわち  $R_4 + S_3$ ）および丸めを通して  $S_4$  を形成する。あらゆる小数值は次の和に累積される（丸める前）、すなわち  $S_1$  は 0 . 1 を有し、 $S_2$  は 0 . 3 を有し、 $S_3$  は 0 . 6 を有し、 $S_4$  は 1 . 0 を有する（丸める前）ことに留意する。丸め器は、和から前の和を差し引くことによって、丸められた割り当て値を形成する。具体的には、 $R_1' = S_1$ （すなわち  $S_1 - 0$ ）、 $R_2' = S_2 - S_1$ 、 $R_3' = S_3 - S_2$ 、および  $R_4' = S_4 - S_3$  である。3 番目の値では、累積された小数值が 0 . 5 以上である場合であるため、切り上げが行われることに留意する。丸めは順序に依存することに留意する。したがって、区画の順序により、どの区画が丸めを受けるかが決まる。たとえば、以下の小数值：0 . 4、0、および 0 . 1 がある場合、0 . 1 を有する 3 番目のアプリケーションは、この累積値は 0 . 5 以上のものであり、かつ小数值 0 . 4 未満であるため、切り上げを受けることになる。区画が 0、0 . 1、および 0 . 4 に順序替えされる場合には、0 . 4 を有する 3 番目のアプリケーションは丸めを受ける。割り当て値が非常に小さい場合を除き、丸めにより、本発明のシステムが大きく乱れることにはならない、すなわち目標を上回る / 下回ることにはならないことに留意する。割り当て値が非常に小さい場合では、小さな値を 1 だけ増加することは、大きな割合の変更を表し、目標を上回る / 下回ることになりうる。たとえば、割り当て値 2 . 1 が 3 に切り上げられるものと想定する。これは、割り当て値よりも 1 4 3 % 大きい値を表す。かかる大きな差は、目標を上回る / 下回ることにつながりうる。

30

40

#### 【 0 0 4 8 】

なお本明細書に図示し説明した例は例示のためのものにすぎず、本発明は他の値でも動

50

作する。

【0049】

さらに、図3に示し図4Aおよび図4Bに示す例と共に例示した割り当て機構は、アプリケーション優先度グループを有する各区画が概ね等しい扱いを受けるように設計される。代替のものも開発可能である。たとえば、それぞれの要求量を受け取る区画の数を最大化しようとするようにPLMをプログラミングすることができる。これにより、同じアプリケーション優先度グループ、特に要求区画がより大きなアプリケーションを有する区画のいくつかが必要になる。その結果、他の区画すなわちより小さな要求区画が満たされる。他の方法では、割り当て量と要求量との差分に比例する量を区画に受け取らせるものがある。所定のアプリケーション優先度レベルに達し、要求資源に対し利用可能な資源が不足することになる場合、差分に比例する量を割り当てることで、各区画が同じ分点に配置される。これにより、区画のいずれも要求する全体量を受け取らず（丸めを受ける）、それぞれの差分によってすべてスケールされる、その結果要求量を受け取る数が最小化される。図3の機構の利点は、区画が、優先度の同じ、または優先度のより低い任意の他の区画（より大きな要求を有する）に影響されないことである。なお区画の要求量を少なくすることにより、それぞれの割り当てが等しくなるまで、等しい優先度の中で多い方の資源区画を減らすことができる。優先度のより高い区画がより多くの資源を要求し始める場合、優先度のより低い区画は資源を失うことになる。しかし、同じ優先度の区画がより多くの資源の要求を開始しこの区画の割り当てが優先度の同じ区画の割り当てよりも小さい場合、この区画が減らしうるのは優先度の同じ区画の資源のみである。したがって、優先度の同じ区画は互いに保護し合う。上述した代替の機構の場合、それぞれの同優先度区画の要求が変化するので、特定の区画の割り当てが影響を受けることになる。

10

20

【0050】

ソフトウェアで実施される場合、本発明の要素は、本質的に、必要なタスクを実行するコードセグメントである。プログラムまたはコードセグメントは、プロセッサ可読媒体に格納するか、あるいは搬送波で具現されるコンピュータデータ信号または搬送波によって変調された信号により伝送媒体を介して伝送することができる。「プロセッサ可読媒体」は、情報を格納または転送することが可能なあらゆる媒体を含むことができる。プロセッサ可読媒体の例としては、電子回路、半導体メモリ素子、ROM、フラッシュメモリ、消去可能ROM（EROM）、フロッピーディスク、コンパクトディスクCD-ROM、光ディスク、ハードディスク、光ファイバ媒体、無線周波数（RF）リンク等が挙げられる。コンピュータデータ信号は、電子ネットワークチャネル、光ファイバ、空気、電磁気、RFリンク等の伝送媒体を介して伝播可能なあらゆる信号を含むことができる。コードセグメントは、インターネット、イントラネット等のコンピュータネットワークを介してダウンロードすることができる。

30

【0051】

図7は、本発明を使用するように適合されたコンピュータシステム700を示す。中央演算処理装置（CPU）701がシステムバス702に接続される。CPU701は、HP PA-8200プロセッサやIntel Pentium IIプロセッサ等任意の汎用CPUであることができる。しかし、本発明は、CPU701が本明細書に記載する本発明の動作をサポートする限り、CPU701のアーキテクチャによって制限されない。バス702はランダムアクセスメモリ（RAM）703に接続され、RAM703はSRAM、DRAM、またはSDRAMであることができる。バス702にはROM704も接続され、ROM704はPROM、EPROM、またはEEPROMであることができる。RAM703およびROM704は、当分野において周知のように、ユーザデータ、システムデータ、およびプログラムを保持する。

40

【0052】

バス702には、入出力（I/O）コントローラカード705、通信アダプタカード711、ユーザインタフェースカード708、およびディスプレイカード709も接続される。I/Oカード705は、ハードドライブ、CDドライブ、フロッピーディスクドライ

50

ブ、テープドライブのうちの1つまたは複数等の記憶装置706をコンピュータシステムに接続する。通信カード711は、コンピュータシステム700をネットワーク712に接続するように適合され、ネットワーク712は、ローカルネットワーク(LAN)、広域ネットワーク(WAN)、イーサネット、またはインターネット網のうちの1つまたは複数であることができる。ユーザインタフェースカード708は、キーボード713およびポインティングデバイス707等ユーザ入力装置をコンピュータシステム700に接続する。ディスプレイカード709はCPU701によって駆動され、表示装置710上の表示を制御する。

【図面の簡単な説明】

【0053】

10

【図1A】従来技術による資源マネージャのブロック図。

【図1B】図1Aのアプリケーションの分割を示すブロック図。

【図2A】複数の区画を使用して動作する本発明の区画負荷マネージャ(PLM)を示すブロック図。

【図2B】図2Aの区画を示すブロック図。

【図3】図2AのPLMの動作のフローチャート。

【図4A】図2AのPLMによる資源の割り当て例を示す図。

【図4B】図2AのPLMによる資源の割り当て例を示す図。

【図5A】図2AのPLMの丸め器の動作を示すブロック図。

【図5B】図2AのPLMの丸め器の動作を示すブロック図。

20

【図5C】図2AのPLMの丸め器の動作を示すブロック図。

【図6】本発明の使用に適合されたコンピュータシステムのブロック図。

【符号の説明】

【0054】

10 プロセス資源マネージャ

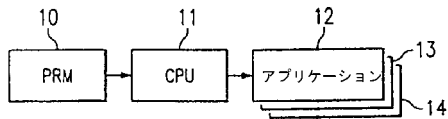
11 CPU

12,13,14 アプリケーション

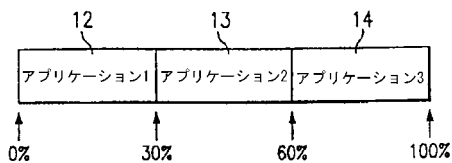
20-1~20-N 作業負荷マネージャ

201 区分負荷マネージャ

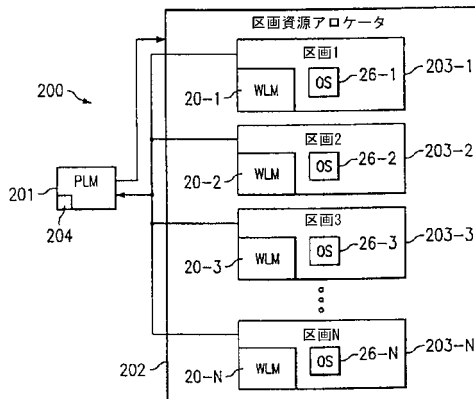
【図1A】



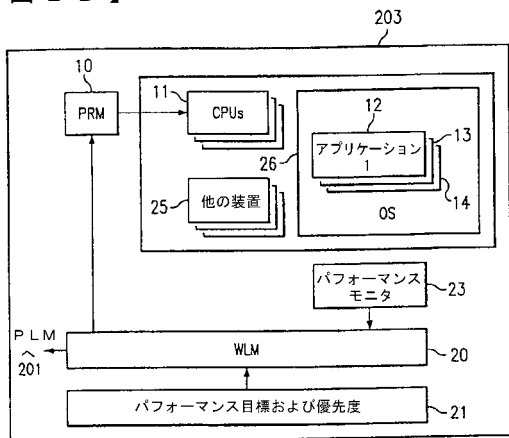
【図1B】



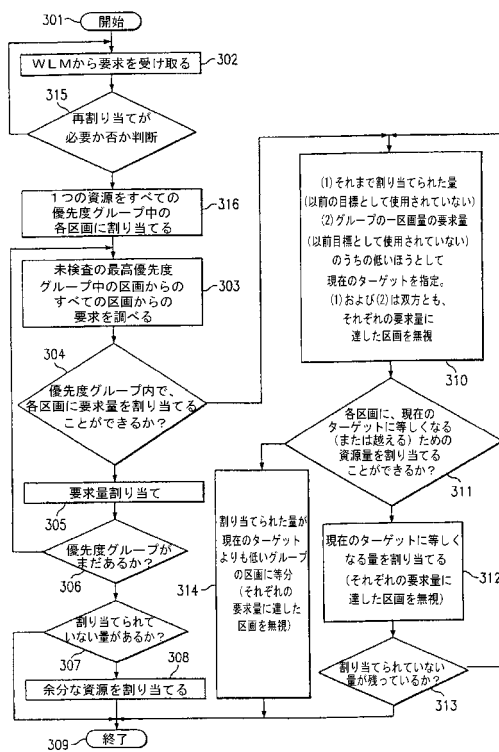
【図2A】



【図2B】



【図3】





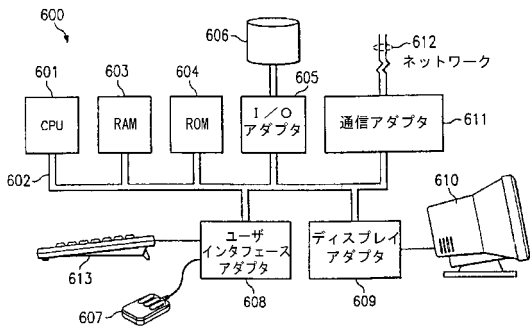
【図 4 A】

区画	要求				19個の資源の累積割り当て						
	優先度				ボックス 316	優先度					
	1	2	3	4		1	2	3	4 (1)	4 (3)	
1	1	1	1	3	1	1	1	1	1	3	4
2	0	3	4	5	1	1	3	4	4	4	4
3	2	4	4	4	1	2	4	4	4	4	4
4	0	0	0	3	1	1	1	1	1	3	4
5	2	3	4	5	1	2	3	4	4	4	4
6	1	1	1	1	1	1	1	1	1	1	1
累積合計	6	12	14	21	6	8	13	15	15	19	
					401	402	403	404	405	406	

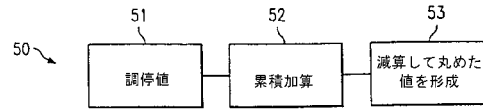
【図 4 B】

区画	要求		19個の資源の累積割り当て								
	優先度		ボックス 316	優先度							
	1	2		1	2 (1)	2 (2)	2 (3)	2 (4)	2 (5)	2 (6)	
1	1	9	1	1	1	2	3	4	5	6.5	
2	1	8	1	1	1	2	3	4	5	6.5	
3	4	5	1	4	4	4	4	4	5	5	
4	1	2	1	1	1	2	2	2	2	2	
5	4	3	1	4	4	4	4	4	4	4	
累積合計	11	27	5	11	11	19	16	18	21	24	
			408	409	410	411	412	413	414	415	

【図 6】



【図 5 A】



【図 5 B】

$$\begin{aligned}
 R1 &= 3.5 & S1 &= R1 + 0 = 3.5 \rightarrow 4 & R1' &= S1 - 0 = 4 \\
 R2 &= 3.5 & S2 &= R1 + R2 = 7.0 \rightarrow 7 & R2' &= S2 - S1 = 3 \\
 R3 &= 3.0 & S3 &= R1 + R2 + R3 = 10.0 \rightarrow 10 & R3' &= S3 - S2 = 3
 \end{aligned}$$

【図 5 C】

$$\begin{aligned}
 R1 &= 10.1 & S1 &= R1 + 0 = 10.1 \rightarrow 10 & R1' &= S1 - 0 = 10 \\
 R2 &= 20.2 & S2 &= R1 + R2 = 30.3 \rightarrow 30 & R2' &= S2 - S1 = 20 \\
 R3 &= 30.3 & S3 &= R1 + R2 + R3 = 60.6 \rightarrow 61 & R3' &= S3 - S2 = 31 \\
 R4 &= 39.4 & S4 &= R1 + R2 + R3 + R4 = 100.0 \rightarrow 100 & R4' &= S4 - S3 = 39
 \end{aligned}$$

---

フロントページの続き

- (72)発明者 クリフォード・エイ・マッカーシー  
アメリカ合衆国75080テキサス州リチャードソン、フォレスト・グローヴ・ドライブ 403
- (72)発明者 トーマス・イー・テュリッチ  
アメリカ合衆国75206テキサス州ダラス、マッコーマス・ブルヴァード 5903
- (72)発明者 スティーヴン・アール・ランドハー  
アメリカ合衆国70806ルイジアナ州バートン・ルージュ、バーガンディ・アヴェニュー 56  
52
- Fターム(参考) 5B098 AA10 GA01 GD02 GD03 GD07 GD14 HH01 HH07 HH08