

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4959845号
(P4959845)

(45) 発行日 平成24年6月27日(2012.6.27)

(24) 登録日 平成24年3月30日(2012.3.30)

(51) Int.Cl.

G06F 9/50 (2006.01)

F 1

G06F 9/46 465D

請求項の数 8 (全 11 頁)

(21) 出願番号 特願2010-547130 (P2010-547130)
 (86) (22) 出願日 平成21年1月28日 (2009.1.28)
 (65) 公表番号 特表2011-513807 (P2011-513807A)
 (43) 公表日 平成23年4月28日 (2011.4.28)
 (86) 國際出願番号 PCT/EP2009/050914
 (87) 國際公開番号 WO2009/106398
 (87) 國際公開日 平成21年9月3日 (2009.9.3)
 審査請求日 平成23年8月1日 (2011.8.1)
 (31) 優先権主張番号 08151917.5
 (32) 優先日 平成20年2月26日 (2008.2.26)
 (33) 優先権主張国 歐州特許庁 (EP)

早期審査対象出願

(73) 特許権者 390009531
 インターナショナル・ビジネス・マシーンズ・コーポレーション
 INTERNATIONAL BUSINESS MACHINES CORPORATION
 アメリカ合衆国10504 ニューヨーク
 州 アーモンク ニュー オーチャード
 ロード
 (74) 代理人 100108501
 弁理士 上野 剛史
 (74) 代理人 100112690
 弁理士 太佐 種一
 (74) 代理人 100091568
 弁理士 市位 嘉宏

最終頁に続く

(54) 【発明の名称】ワーカロード・マネージャにおいてディスパッチャのシェアを計算する方法、コンピュータ・プログラム及びデータ処理システム

(57) 【特許請求の範囲】

【請求項1】

ワーカロード・マネージャ(101)において、ディスパッチャがアービトレータから受け取るワーカロードのシェアを示す、ディスパッチャのシェア(D)を計算する方法であって、

前記ワーカロード・マネージャ(101)はアービトレータ(102)に結合され、前記アービトレータ(102)は複数のシステム(117~119)に結合され、各システムはディスパッチャ(103、104又は105)を備え、各ディスパッチャ(103、104又は105)は複数の実行ユニット(106~108、109~111又は112~114)に結合され、前記アービトレータ(102)はワーカロード項目(115)のフローを受け取り且つこれを前記ディスパッチャ(103~105)に配分するのに適合しており、前記実行ユニット(106~114)は前記ワーカロード項目(115)を実行するのに適合しており、前記ワーカロード項目(115)は少なくとも第1~第3のワーカロード・タイプを有しており、

前記方法は、

前記複数のシステム(117~119)にわたる各ワーカロード・タイプ用のサービス単位の合計値(W)を獲得するステップを有し、前記サービス単位は時間間隔でCPU消費量を示す値であり、

前記複数のシステム(117~119)の各システム上の各ワーカロード・タイプ用のキャパシティ値(cap)を獲得するステップを有し、前記キャパシティ値(cap)は

前記時間間隔でシステムが実行することができる最大サービス単位を示し、

各システム上の各ワークロード・タイプの前記キャパシティ値(*c a p*)を各ワークロード・タイプの前記サービス単位の合計値(*W*)で除算することにより、各システムのディスパッチャの相対シェア(*R*)を計算するとともに、各システムの当該ディスパッチャの相対シェア(*R*)のうち最小値を獲得するステップを有し、

各システム上の各ワークロード・タイプ用の前記ワークロード項目(115)のキューレンジ(*q*)を各システム上の各ワークロード・タイプの前記キャパシティ値(*c a p*)で除算することにより、各システム用の相対キューレンジ(*V*)を計算するステップを有し、

前記最小値と(1 + 最大の前記相対キューレンジ(*V*))の逆数値との乗算により、各システム用の前記ディスパッチャのシェア(*D*)を計算するステップを有する方法。 10

【請求項 2】

前記相対キューレンジ(*V*)の前記第1の関数(427)は、前記相対キューレンジ(*V*)の最大値の逆数値を計算する、請求項1に記載の方法。

【請求項 3】

前記算術演算は乗算である、請求項1又は請求項2に記載の方法。

【請求項 4】

前記ディスパッチャのシェア(*D*)は、当該ディスパッチャのシェア(*D*)を全てのシステムの全てのディスパッチャの和で除算することにより正規化される、請求項1ないし請求項3の何れか1項に記載の方法。 20

【請求項 5】

前記最小値は、当該最小値を全てのシステムの全ての最小値の和で除算することにより正規化される、請求項1ないし請求項4の何れか1項に記載の方法。

【請求項 6】

前記アービトレータは、前記ワークロード項目(115)のフローを前記ディスパッチャ(103 、 104 又は 105)に配分する、請求項1ないし請求項5の何れか1項に記載の方法。 20

【請求項 7】

請求項1ないし請求項6の何れか1項に記載の方法の各ステップをコンピュータに実行させるためのコンピュータ・プログラム。

【請求項 8】

ワークロード・マネージャ(101)において、ディスパッチャがアービトレータから受け取るワークロードのシェアを示す、ディスパッチャのシェア(*D*)を計算するためのデータ処理システムであって、 30

前記ワークロード・マネージャ(101)はアービトレータ(102)に結合され、前記アービトレータ(102)は複数のシステム(117 ~ 119)に結合され、各システムはディスパッチャ(103 、 104 又は 105)を備え、各ディスパッチャ(103 、 104 又は 105)は複数の実行ユニット(106 ~ 108 、 109 ~ 111 又は 112 ~ 114)に結合され、前記アービトレータ(102)はワークロード項目(115)のフローを受け取り且つこれを前記ディスパッチャ(103 ~ 105)に配分するのに適合しており、前記実行ユニット(106 ~ 114)は前記ワークロード項目(115)を実行するのに適合しており、前記ワークロード項目(115)は少なくとも第1 ~ 第3のワークロード・タイプを有しており。 40

前記データ処理システムは、

前記複数のシステム(117 ~ 119)にわたる各ワークロード・タイプ用のサービス単位の合計値(*W*)を獲得するためのコンポーネントを備え、前記サービス単位は時間間隔で C P U 消費量を示す値であり、

前記複数のシステム(117 ~ 119)の各システム上の各ワークロード・タイプ用のキャパシティ値(*c a p*)を獲得するためのコンポーネントを備え、前記キャパシティ値(*c a p*)は前記時間間隔でシステムが実行することができる最大サービス単位を示し、

各システム上の各ワークロード・タイプの前記キャパシティ値(*c a p*)を各ワークロ

50

ード・タイプの前記サービス単位の合計値 (W) で除算することにより、各システムのディスパッチャの相対シェア (R) を計算するとともに、各システムの当該ディスパッチャの相対シェア (R) のうち最小値を獲得するためのコンポーネントを備え、

各システム上の各ワークロード・タイプ用の前記ワークロード項目(115)のキューランク(q)を各システム上の各ワークロード・タイプの前記キャパシティ値(cap)で除算することにより、各システム用の相対キューランク(v)を計算するためのコンポーネントを備え、

前記最小値と(1+最大の前記相対キューラン(Ⅴ))の逆数値との乗算により、各システム用の前記ディスパッチャのシェア(D)を計算するためのコンポーネントを備える、データ処理システム。

10

【発明の詳細な説明】

【技術分野】

[0 0 0 1]

本発明は、ワークロード・マネージャにおいてルーティング・ワークロードを計算する方法及びコンピュータ・プログラムに係る。

【背景技术】

[0 0 0 3]

メインフレームは、重要なアプリケーションを実行し且つ金融取引処理のような大量なデータを処理するために、主として大組織によって使用されるコンピュータである。メインフレームは、信頼性及び安全なシステムを提供するために、高度の冗長性を有している。メインフレームは、複数のオペレーティング・システムを実行又はホストすることができるから、現用中の多数の小型サーバをメインフレームに置き換えると、管理負担を軽減させ且つ改良されたスケーラビリティを提供することができる。最近のメインフレームは、IBM社のz Series及びシステムz9サーバを含む。

30

【 0 0 0 3 】

並列システムプレックスは、z/OSを使用して、単一のシステム・イメージとして協調して動作する、複数のIBMメインフレームのクラスタである。システムプレックスは、並行処理を使用し且つ完全なデータ保全性を維持しつつ、複数のシステムにわたる読み取り/書き込みデータの共有を可能にすることにより、最大32個のシステムをクラスタ化し且つこれらのシステムにわたり一のワークロードを共有する。このワークロードについては、これを一のシステム内の複数の個別プロセッサにわたり動的に配分(distribute)したり、使用可能な資源を有するクラスタ内の任意のシステムに配分することができる。また、ワークロード平衡化は、クリティカルな応答レベルを維持しつつ、並列システムプレックス・クラスタにわたり種々のアプリケーションを実行することを可能にする。もし、ワークロード平衡化又はワークロード・ルーティングが正確に行われなければ、システム内で過負荷が生じることがある。

30

【発明の概要】

【発明が解決しようとする課題】

[0 0 0 4]

従って、当分野では、ワーカロード・マネージャにおいてディスパッチャのシェアを効率的に計算する方法、並びに当該方法を実行するのに適したワーカロード・マネージャ及びコンピュータ・プログラムが要請されている。

10

【課題を解決するための手段】

卷之三

50

ド項目は少なくとも第1～第3のワークロード・タイプを有する。前記方法は、メモリから前記複数のシステムにわたる各ワークロード・タイプ用のサービス単位の合計値(W)を読み取るステップを有し、前記サービス単位は一の時間インターバル中のCPU消費量を測定するための値であり、前記メモリから各システム上の各ワークロード・タイプ用のキャパシティ(cap)値を読み取るステップをさらに有し、前記キャパシティ値は一のシステムが実行することができる最大サービス単位を示す。

【 0006 】

さらに、前記方法は、各システム上の各ワークロード・タイプの前記キャパシティ値を各ワークロード・タイプの前記サービス単位の合計値で除算することにより、各システムのディスパッチャの相対シェア(R)を計算するとともに、各システムの当該ディスパッチャの相対シェア(R)のうち最小値を獲得するステップと、各システム上の各ワークロード・タイプ用の前記ワークロード項目のキュー長(q)を各システム上の各ワークロード・タイプの前記キャパシティ値で除算することにより、各システム用の相対キュー長(V)を計算するステップと、算術演算を使用して、前記最小値と前記相対キュー長(V)の第1の関数を組み合わせることにより、各システム用の前記ディスパッチャのシェア(D)を計算するステップとを有する。

【 0007 】

前記ディスパッチャのシェアは、全てのルーティングに対するディスパッチャのシェア、すなわち全てのディスパッチャがアービトレイタから受け取るワークロードのシェアを指す。前記ディスパッチャの相対シェアは、最小の相対実行キャパシティのディスパッチャの相対シェアを指す。

【 0008 】

前記方法の前記相対キュー長の前記第1の関数は、前記相対キュー長の最大値の逆数値を計算することを含む。前記算術演算は乗算である。前記第1～第3のワークロード・タイプは、general CPU(C P)、z Application Assist Processor(z A A P)及び z9 Integrated Information Processor(z I I P)ワークロード・タイプである。前記ディスパッチャのシェアは、当該ディスパッチャのシェアを全てのシステムの全てのディスパッチャの和で除算することにより正規化される。

【 0009 】

前記方法の前記最小値は、当該最小値を全てのシステムの全ての最小値の和で除算することにより正規化される。前記アービトレイタは、前記ワークロード項目のフローを他のシステム内の他のディスパッチャに配分する。

【 0010 】

他の側面では、本発明は、前記方法の各ステップをコンピュータに実行させるためのコンピュータ・プログラムに係る。

【 0011 】

他の側面では、本発明は、ワークロード・マネージャにおいてディスパッチャのシェアを計算するためのデータ処理システムに係る。

【発明の効果】

【 0012 】

実施形態の利点のうちの1つは、シスプレックスの合計キャパシティが依然として全てのワークロード・タイプ用の合計ワークロード・デマンドを超過する限り、キャパシティ不足又は一の実行ユニットの停止(outage)の結果として、ディスパッチャ及び実行ユニット内のキューが無限に成長しないということである。これらの場合、各システムに対する全てのルーティングのディスパッチャのシェアは、任意のプロセッサ上のそのキャパシティに相対的なキュー長及びそのキャパシティに対する一のワークロード・タイプのサービス単位の合計値の相対値に依存する。1つのシステム上の一の実行ユニットが停止する場合は、他のシステム内にある諸資源のオーバーフロー又は輻輳に帰着することなく、ワークロードを当該他のシステムに配分することにより、当該実行ユニットを補償することが可能である。他の利点は、ディスパッチャのシェアの計算が初期のワークロード配分か

10

20

30

40

50

ら独立していること及びアルゴリズムが最適のワークロード配分に集中することである。

【図面の簡単な説明】

【0013】

【図1】本発明の実施形態に従った、シスプレックス100のブロック図である。

【図2】本発明の実施形態に従った、シスプレックス200を形成するように相互に接続された複数のシステムのブロック図である。

【図3】本発明の実施形態に従った、ディスパッチャのシェアを計算する方法のフローチャートである。

【図4】本発明の実施形態に従った、ワークロード・マネージャの計算のブロック図である。

10

【発明を実施するための形態】

【0014】

図1のシスプレックス100は、第1のシステム118を備え、その第1のワークロード・マネージャ101は、アビトレータ102に結合される。アビトレータ102は、第1～第3のディスパッチャ103～105に結合される。第1のワークロード・マネージャ101は、他の2つのワークロード・マネージャ120及び121に結合される。第1のディスパッチャ103は、第2のシステム117内に設けられ、3つの実行ユニット106～108に結合される。第2のディスパッチャ104は、第1のシステム118内に設けられ、3つの実行ユニット109～111に結合される。第3のディスパッチャ105は、第3のシステム119内に設けられ、他の3つの実行ユニット112～114に結合される。メモリ116は、ワークロード・マネージャ101に結合される。

20

【0015】

アビトレータ102は、入来する複数のワークロード項目115を受け取り、それらをディスパッチャ103～105に送付する。ディスパッチャ103～105は、当該ワークロード項目115のワークロード・タイプを決定し、それらを専門化された実行ユニット106～114に送付する。前記複数のワークロード項目115のキュー長(queue length)は、各ディスパッチャごと及び各実行ユニットごとに測定することができる。例えば、ディスパッチャ103は、5ワークロード項目のキュー長(q_{D1})を有し、実行ユニット106は、3ワークロード項目のキュー長($q_{E1,1}$)を有する。全てのディスパッチャ又は全ての実行ユニットについてキュー長を測定する必要はない。キュー長は、アビトレータ102の背後にあるルーティングの少なくとも1つのステージにおいて測定しなければならない。これは、本発明の実施形態の唯1つの要件である。

30

【0016】

各ワークロード項目は、一の時間インターバル中に一の実行ユニットの異なるCPU消費量を必要とするが、その消費量は、アビトレータ102には前もって知られていない。従って、ディスパッチャ103～105に対する複数のワークロード項目115の割り当ては、各ワークロード項目115のサイズ又はそのCPU消費量に依存しない。各ワークロード項目115のワークロード・タイプは、ディスパッチャ103～105によって識別され、ワークロード項目115のサイズは、ディスパッチャ103～105又は実行ユニット106～114によって識別される。前述のように、アビトレータ102は、ワークロード項目115のワークロード・タイプ及びワークロード項目115のサイズを前もって知らないから、間違ったアルゴリズムが一のキューを無限に成長させことがある。全てのシステム117～119のワークロード・マネージャ101、120及び121は、相互に結合され且つ定期的に対話することにより、当該システムのキャパシティ及びワークロード値を通信する。メモリ116は、前記複数のシステムにわたる全てのワークロード・タイプのサービス単位の合計値及び各システムの全てのワークロード・タイプのキャパシティ値を格納する。

40

【0017】

図2に例示するシスプレックス200は、相互に接続された第1～第4のシステム201～204によって形成される。システム201～204は、ワークロード・マネージャ

50

205～208を含み、これらのワークロード・マネージャ205～208の全ては、互いに結合されている。ワークロード・マネージャ205及び206は、アービトレータ209及び210に結合される。代替的に、シスプレックス200内に1つのアービトレータを設けることで十分である。アービトレータ209及び210は、複数のディスパッチャ213～216に結合され、一方、ディスパッチャ213～216は、実行ユニット217～220にそれぞれ結合される。全てのシステム201～204が一のアービトレータを必要とするとは限らない。というのは、アービトレータは、シスプレックス200内の任意のシステムにワークロードを送付することができるからである。

【0018】

ワークロード・マネージャ205～208は、それぞれルーティング・アルゴリズムを処理する。このルーティング・アルゴリズムは、一のディスパッチャが受け取るべきワークロード項目の相対量のディスパッチャのシェアを計算する。アービトレータ209及び210は、ワークロード項目のワークロード・タイプ及びワークロード項目のサイズを知らない。このため、アービトレータ209及び210は、ワークロード・マネージャ205～208からルーティング推奨(recommendations)を受け取り、ルーティング・アルゴリズムの結果に従ってワークロード項目を配分する。

【0019】

キュー・マネージャとも称するディスパッチャ213～216は、アービトレータ209及び210からワークロード項目を受け取り、サーバとも称する実行ユニット217～220によって取り出されるまで、それらのワークロード項目を待ち行列化する。実行ユニット217～220は、ワークロード項目を実行し、ワークロード・タイプを読み取り、どのプロセッサがワークロード項目を処理することができるかを決定する。ワークロード・タイプは、general CPU (CP)、z Application Assist Processor (zAAP) 及び z9 Integrated Information Processor (zIIP) ワークロード・タイプを含む、3つ以上のタイプとができる。各ワークロード・タイプごとに、異なるプロセッサが使用される。ワークロード・マネージャ205～208は、全てのシステム201～204にわたってディスパッチャ・ステータスに関係する情報を受け取るように、相互に結合されている。

【0020】

図3は、ワークロード・マネージャにおいてディスパッチャのシェア(D)を計算する方法のフローチャート300を示す。第1のステップ301は、進行中の時間インターバルの間に、複数のシステムにわたる各ワークロード・タイプ用のサービス単位の合計値(W)を獲得する。サービス単位は、一の時間インターバル中のCPU消費量を示す値である。サービス単位の合計値(W)は、第1のワークロード・タイプ用の第1のサービス単位値と、第2のワークロード・タイプ用の第2のサービス単位値と、第3のワークロード・タイプ用の第3のサービス単位値とを含む。システム内でさらなるワークロード・タイプが使用可能である場合は、これに対応するサービス単位の合計値(W)が獲得される。

【0021】

第2のステップ302は、複数のシステムの各システム上の第1～第3のワークロード・タイプを含む、各ワークロード・タイプ用の一のキャパシティ値(cap)を獲得する。キャパシティ値(cap)は、一の時間インターバル中に一のシステムが実行することができるサービス単位の最大量を示す。各ワークロード・タイプごと及び各システムごとに、異なるキャパシティ値(cap)が獲得される。従って、各システムは、各ワークロード・タイプ用の一のキャパシティ値(cap)を含むであろう。

【0022】

第3のステップ303は、ディスパッチャの相対シェア(R)を計算する。この相対シェア(R)は、各システム上の各ワークロード・タイプ用のキャパシティ値(cap)を、各ワークロード・タイプ用のサービス単位の合計値(W)で除算することにより獲得される。各システム上の各ワークロード・タイプ用に獲得されたキャパシティ値(cap)は、同じワークロード・タイプのサービス単位の合計値(W)で除算され、そしてその最

10

20

30

40

50

小値は、各システムのディスパッチャの相対シェアに対応する。第3のステップ303では、各システムごとに単一の値が獲得される。

【0023】

第4のステップ304は、各システムからの相対キュー長(V)を計算する。この相対キュー長(V)は、各システム上の各ワークロード・タイプ用のワークロード項目のキュー長(q)を、各システム上の各ワークロード・タイプのキャパシティ値(cap)で除算することにより獲得される。各システムは、各ワークロード・タイプ用の一の値を獲得する。

【0024】

第5のステップ305は、算術演算を使用して、前記最小値及び前記相対キュー長(V)の第1の関数を組み合わせることにより、各システム用のディスパッチャのシェア(D)を計算する。

【0025】

図4は、ワークロード項目のフローを複数のディスパッチャにわたって配分するための、ディスパッチャのシェア(D)の計算を例示する。第1のテーブル400は、第1～第3のシステム401～403と、第1～第3のワークロード・タイプ404～406と、サービス単位の合計値(W)407と、各システム上の複数のキャパシティ値(cap)、すなわち第1のシステム401上の複数の第1のキャパシティ値408、第2のシステム402上の複数の第2のキャパシティ値409及び第3のシステム403上の複数の第3のキャパシティ値410とを含む。

【0026】

第2のテーブル420は、各システム上及び各ワークロード・タイプの複数のキュー長(q)421～423と、各システム上及び各ワークロード・タイプ用の計算された相対キュー長(V)424～426と、各システム上及び各ワークロード・タイプ用の相対キュー長の関数427とを含む。

【0027】

ワークロード・マネージャにおいてディスパッチャのシェア(D)を計算するための第1のステップのうちの1つは、全てのシステム401～403にわたる各ワークロード・タイプのサービス単位の合計値(W)を獲得することを含む。この例では、第1のシステム401上の第1のワークロード・タイプ用のサービス単位の合計値は「300」であり、第2のワークロード・タイプ用のサービス単位の合計値は「500」であり、第3のワークロード・タイプ用のサービス単位の合計値は「10」である。

【0028】

他のステップは、各システムごと及び各ワークロード・タイプごとのキャパシティ値(cap)を獲得することを含む。例えば、第1のシステム401上では、第1のワークロード・タイプ用のキャパシティ値は「90」であり、第2のワークロード・タイプ用のキャパシティ値は「100」であり、第3のワークロード・タイプ用のキャパシティ値は「10」である。第2のシステム402上では、第1のワークロード・タイプ用のキャパシティ値は「200」であり、第2のワークロード・タイプ用のキャパシティ値は「400」であり、第3のワークロード・タイプ用のキャパシティ値は「100」である。全てのシステムについてキャパシティ値が獲得された後、各システム上で各ワークロード・タイプ用のディスパッチャの相対シェア(R)を計算することができる。

【0029】

第1のシステム401上では、システム401が処理することができる第1のワークロードは、第1のワークロード・タイプ用のキャパシティ値「90」を、第1のワークロード・タイプのサービス単位の合計値「300」で除算し、その結果「0.3」を導くことにより、獲得することができる。第2のワークロード・タイプの第2のキャパシティ値「100」を、第2のワークロード・タイプのサービス単位の合計値「500」で除算すると、システム401が処理することができる第2のワークロードは「0.2」となる。この値は、システム401が処理することができる実際のワークロードを、パーセンテージ

10

20

30

40

50

で表したものである。第3のワークロード・タイプについて同じプロセスを繰り返すと、その結果は「1」になる。また、他のシステム402及び403にわたって同じプロセスを繰り返すと、各システムごとに3つの異なる値が獲得される。

【0030】

ディスパッチャのシェアを計算する前記方法の他のステップは、以前に獲得したディスパッチャの相対シェア(R)の最小値を計算することを含む。例えば、第1のシステム401上では、ディスパッチャの相対シェア(R)として、「0.3」、「0.2」及び「1」の3つの値がある。これらの3つの値の最小値は、「0.2」である。第2のシステム402については、ディスパッチャの相対シェア(R)の最小値は「0.6667」であり、第3のシステム403については、ディスパッチャの相対シェア(R)の最小値は「1」である。10

【0031】

第2のテーブル420は、ディスパッチャのシェア(D)を計算する前記方法の諸ステップを例示する。すなわち、これらのステップは、キューランク(q)421～423を特定のワークロード・タイプ用のキャパシティ値(cap)408～410で除算することにより、相対キューランク(V)424～426を計算することを含む。第1のシステム401上では、第1のワークロード・タイプ用のキューランク421は「125」であり、第2のワークロード・タイプ用のキューランク421は「89」であり、第3のワークロード・タイプ用のキューランク421は「67」である。第1のシステム401上では、第1の相対キューランク424は、「125」を「90」で除算することにより「1.388」として獲得され、第2の相対キューランク424は、「89」を「100」で除算することにより「0.89」として獲得される。全てのシステム401～403にわたって全てのワークロード・タイプの全てのキューランクについて同じプロセスを繰り返すと、第1のシステム401の第3の相対キューランク424は「6.7」として獲得される。20

【0032】

さらに、第2のテーブル420は、相対キューランク(V)424～426の第1の関数427を含む。この関数は、「1+最大の相対キューランク」の逆数値を獲得するためのものである。第1のシステム401上では、最大の相対キューランクは「6.7」であるから、その値+1の逆数値は「0.1299」である。第2のシステム402上では、最大の相対キューランクは「3.655」であるから、その値+1の逆数値は「0.2148」である。第3のシステム403上では、最大の相対キューランクは「4.3564」であるから、その値+1の逆数値は「0.1867」である。これらの全ての値が計算されると、ディスパッチャのシェア(D)を獲得するために、各システムのディスパッチャの相対シェア(R)の最小値と、各システムの第1の関数値とが乗算される。従って、第1のシステム401については、「0.2」と「0.1299」とを乗算することにより、「0.02598」が獲得される。第2のシステム402については、「0.6667」と「0.2148」とを乗算することにより、「0.1432」が獲得される。30

【0033】

最後に、第3のシステム403については、「1」と「0.1867」とを乗算することにより、「0.1867」が獲得される。これらの3つの値は、各システムごとのディスパッチャのシェアを示すから、アービトレータは、各ディスパッチャごとに異なる量のワークロード単位を送付することができる。その結果、全てのディスパッチャ及び全てのシステムにわたって、ワークロード項目の最良の可能な配分を獲得することができる。予め定められた、一定の時間インターバルの後、ワークロードW、キャパシティ及びキューランクにおける任意の変化を考慮に入れて、前記計算を繰り返すことにより、ディスパッチャのシェア(D)を新しい最適値に調整することができる。40

【0034】

本発明は、完全なハードウェア実施形態、完全なソフトウェア実施形態又はハードウェア及びソフトウェア要素の両方を含む実施形態の形式を取ることができる。好ましい実施形態では、本発明は、(ファームウェア、常駐ソフトウェア、マイクロコード等を含む)50

ソフトウェアの形式で実装される。

【0035】

さらに、本発明は、コンピュータ又は任意の命令実行システムによって或いはそれらに関連して使用するためのプログラム・コードを提供するコンピュータ使用可能媒体又はコンピュータ可読媒体からアクセス可能なコンピュータ・プログラムの形式を取ることができる。この点に関連して、コンピュータ使用可能媒体又はコンピュータ可読媒体は、命令実行システム、装置又はデバイスによってあるいはそれらに関連して使用するためのプログラムを保持し、格納し、通信し、伝播し又は搬送することができる任意の装置とすることができる。

【0036】

かかる媒体は、電子的、磁気的、光学的、電磁的、赤外線、又は半導体システム（装置又はデバイス）或いは伝播媒体とすることができます。コンピュータ可読媒体の例は、半導体又は固体メモリ、磁気テープ、取り外し可能なフレキシブル・ディスク、ランダム・アクセス・メモリ（RAM）、読み取り専用メモリ（ROM）、固定磁気ディスク及び光ディスクを含む。光ディスクの現在の例は、コンパクト・ディスク・リード・オンリ・メモリ（CD-ROM）、コンパクト・ディスク・リード/ライト（CD-R/W）及びDVDを含む。

【0037】

プログラム・コードを格納又は実行もしくはこれらの両方を行うための適切なデータ処理システムは、システム・バスを通してメモリ要素に直接的に又は間接的に結合された少なくとも1つのプロセッサを含むであろう。かかるメモリ要素は、プログラム・コードの実際の実行中に使用されるローカル・メモリと、大容量記憶装置、及び実行中にコードが大容量記憶装置から検索されなければならない回数を減らすために、プログラム・コードの一時的記憶装置を提供するキャッシュ・メモリとを含むことができる。

【0038】

I/Oデバイス（キーボード、ディスプレイ、ポインティング・デバイス等を含む）は、直接的に又はI/Oコントローラを介してシステムに結合することができます。また、ネットワーク・アダプタをデータ処理システムに結合すると、当該データ処理システムを、プライベート・ネットワーク又は公衆ネットワークを介して、他のデータ処理システム、リモート・プリンタ、ストレージ装置等に結合することができる。モデム、ケーブル・モデム及びイーサネット（登録商標）カードは、現時点で使用可能なタイプのネットワーク・アダプタである。

【0039】

以上、本発明の特定の実施形態を説明したが、これらの実施形態において種々の変更及び修正を施しうることは明らかである。本発明の範囲は、各請求項の記載によって定義されるものである。

【符号の説明】

【0040】

101、120、121…ワークロード・マネージャ

102…アービトレータ

103～105…ディスパッチャ

106～114…実行ユニット

115…ワークロード項目

200…システム

201～204…システム

205～208…ワークロード・マネージャ

209～212…アービトレータ

213～216…ディスパッチャ

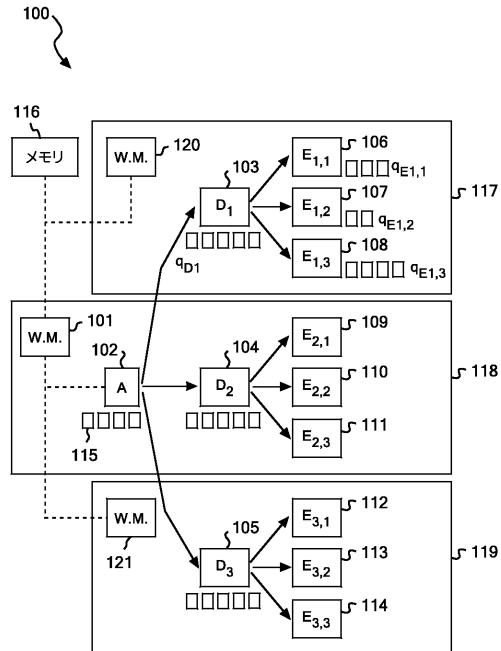
10

20

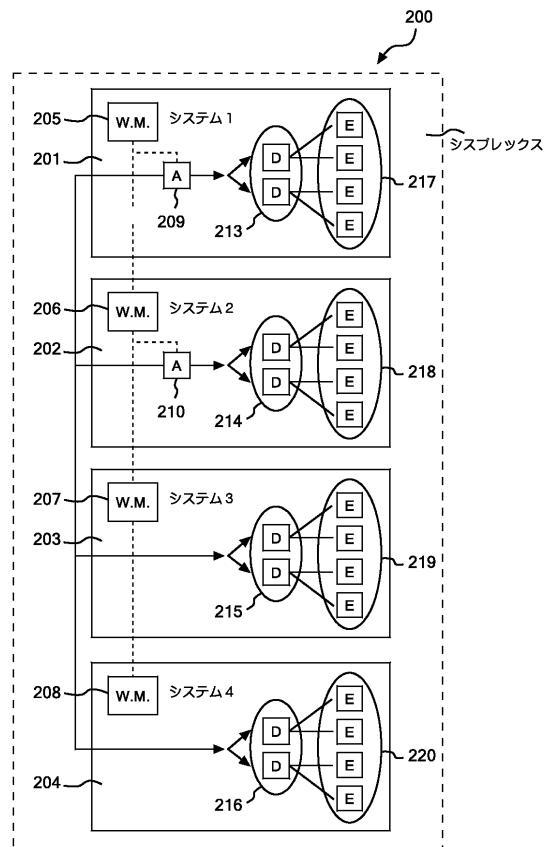
30

40

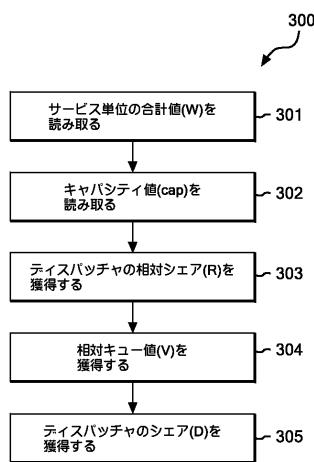
【図1】



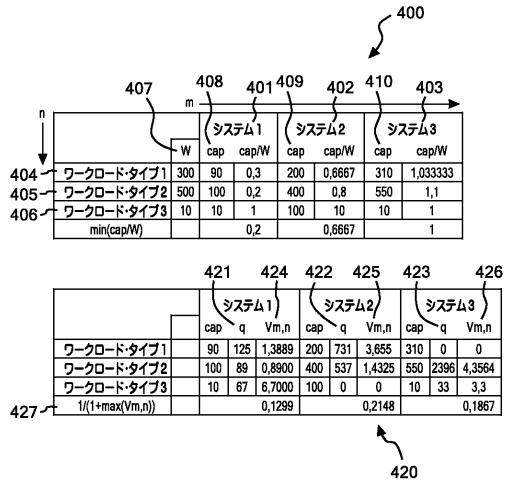
【図2】



【図3】



【図4】



フロントページの続き

(72)発明者 ノイバウアー、マーク
 ドイツ連邦共和国 71034 ベーブリングン シュールシュトラーセ12/1
(72)発明者 ファウペル、ロベルト
 ドイツ連邦共和国 72108 ロッテンブルク ヴァインガルテンシュトラーセ42
(72)発明者 シュリンクマン、ヴェレーナ
 ドイツ連邦共和国 45731 ヴァルトローブ エーゲルメア 48

審査官 井上 宏一

(56)参考文献 特開2006-48680 (JP, A)
 特開2005-71031 (JP, A)
 特開2000-268012 (JP, A)
 特開平6-243112 (JP, A)
 特開2000-259591 (JP, A)
 特開2004-30663 (JP, A)
 特開2008-47126 (JP, A)
 特開平4-318655 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 9/46 -9/54