

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4557949号
(P4557949)

(45) 発行日 平成22年10月6日(2010.10.6)

(24) 登録日 平成22年7月30日(2010.7.30)

(51) Int.Cl. F I
G06F 9/50 (2006.01) G06F 9/46 462A

請求項の数 10 (全 31 頁)

(21) 出願番号	特願2006-285397 (P2006-285397)	(73) 特許権者	000005223 富士通株式会社
(22) 出願日	平成18年10月19日(2006.10.19)		神奈川県川崎市中原区上小田中4丁目1番1号
(65) 公開番号	特開2007-305101 (P2007-305101A)		
(43) 公開日	平成19年11月22日(2007.11.22)	(74) 代理人	100104190 弁理士 酒井 昭徳
審査請求日	平成20年11月17日(2008.11.17)		
(31) 優先権主張番号	特願2006-107631 (P2006-107631)	(72) 発明者	小橋 博道 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
(32) 優先日	平成18年4月10日(2006.4.10)		
(33) 優先権主張国	日本国(JP)	(72) 発明者	繁田 聡一 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	上田 晴康 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】 資源ブローカリングプログラム、該プログラムを記録した記録媒体、資源ブローカリング装置、および資源ブローカリング方法

(57) 【特許請求の範囲】

【請求項1】

各資源ノードに割り当てられた各サービスに応じた優先度を当該各資源ノードに対応付けて設定する第1の設定工程と、

前記各資源ノードによる前記各サービスの処理時間の経過に応じて、前記第1の設定工程によって当該各資源ノードについて設定された優先度を、より優先度の高いものに更新する第2の設定工程と、

新たなサービスを割り当てる空きの資源ノードが存在しない場合に、サービスが既に割り当てられた第1の資源ノードと第2の資源ノードのうち、前記第2の設定工程によって更新された優先度がより低い資源ノードに割り当てられているサービスを当該新たなサービスに切り替える切替工程と、

をコンピュータに実行させることを特徴とする資源ブローカリングプログラム。

【請求項2】

前記第2の設定工程は、

前記第1の設定工程によって当該各資源ノードについて設定された優先度を、当該各資源ノードの処理性能に応じて更新することを特徴とする請求項1に記載の資源ブローカリングプログラム。

【請求項3】

前記第2の設定工程は、

前記第1の設定工程によって当該各資源ノードについて設定された優先度を、当該各資

源ノードの使用頻度に応じて大きくなるように更新することを特徴とする請求項 1 または 2 に記載の資源ブローカリングプログラム。

【請求項 4】

前記第 2 の設定工程は、

前記第 1 の設定工程によって当該各資源ノードについて設定された優先度を、当該各資源ノードの設置位置に応じて更新することを特徴とする請求項 1 ~ 3 のいずれか一つに記載の資源ブローカリングプログラム。

【請求項 5】

前記第 2 の設定工程は、

前記第 1 の設定工程によって当該各資源ノードについて設定された優先度を、指定入力された実行時間内における当該各資源ノードの中断または終了の可否に応じて更新することを特徴とする請求項 1 ~ 4 のいずれか一つに記載の資源ブローカリングプログラム。

10

【請求項 6】

前記第 2 の設定工程は、

前記各サービスに関するジョブの集合のうち実行が完了したジョブの割合に応じて、当該各サービスが割り当てられた前記各資源ノードのうち前記ジョブの実行が完了していない資源ノードの優先度を、より優先度の高いものに更新することを特徴とする請求項 1 ~ 5 のいずれか一つに記載の資源ブローカリングプログラム。

【請求項 7】

前記第 2 の設定工程は、

前記第 1 の設定工程によって当該各資源ノードについて設定された優先度を、当該各資源ノードの切替回数に応じて大きくなるように更新することを特徴とする請求項 1 ~ 6 のいずれか一つに記載の資源ブローカリングプログラム。

20

【請求項 8】

前記切替工程は、

前記新たなサービスを割り当てる空きの資源ノードが存在しない場合に、サービスが既に割り当てられた複数の資源ノードのうち、前記第 2 の設定工程によって更新された優先度が最低の資源ノードに割り当てられているサービスを当該新たなサービスに切り替えることを特徴とする請求項 1 ~ 7 のいずれか一つに記載の資源ブローカリングプログラム。

【請求項 9】

各資源ノードに割り当てられた各サービスに応じた優先度を当該各資源ノードに対応付けて設定する第 1 の設定手段と、

前記各資源ノードによる前記各サービスの処理時間の経過に応じて、前記第 1 の設定手段によって当該各資源ノードについて設定された優先度を、より優先度の高いものに更新する第 2 の設定手段と、

新たなサービスを割り当てる空きの資源ノードが存在しない場合に、サービスが既に割り当てられた第 1 の資源ノードと第 2 の資源ノードのうち、前記第 2 の設定手段によって更新された優先度がより低い資源ノードに割り当てられているサービスを当該新たなサービスに切り替える切替手段と、

30

を備えることを特徴とする資源ブローカリング装置。

40

【請求項 10】

各資源ノードに割り当てられた各サービスに応じた優先度を当該各資源ノードに対応付けて設定する第 1 の設定工程と、

前記各資源ノードによる前記各サービスの処理時間の経過に応じて、前記第 1 の設定工程によって当該各資源ノードについて設定された優先度を、より優先度の高いものに更新する第 2 の設定工程と、

新たなサービスを割り当てる空きの資源ノードが存在しない場合に、サービスが既に割り当てられた第 1 の資源ノードと第 2 の資源ノードのうち、前記第 2 の設定工程によって更新された優先度がより低い資源ノードに割り当てられているサービスを当該新たなサービスに切り替える切替工程と、

50

を含んだことを特徴とする資源ブローカリング方法。

【発明の詳細な説明】

【技術分野】

【0001】

この発明は、複数のサービス間で利用される資源ノードをブローカリングする資源ブローカリングプログラム、該プログラムを記録した記録媒体、資源ブローカリング装置、および資源ブローカリング方法に関する。

【背景技術】

【0002】

従来、システムの構成に柔軟性がないために、処理能力の増強に多額の追加投資が必要になる場合がある。例えば、ビジネスの状況の変化とともに必要なピーク性能が高くなり、追加の計算資源を余儀なくされている。逆に突発的な負荷上昇を見越して予め大量の予備のリソースを確保することもあるが、多くの場合無駄遣いになっている。

【0003】

【非特許文献1】Systemwalker Resource Coordinator：富士通、[online]、[平成18年4月5日検索]、インターネット<URL：<http://systemwalker.fujitsu.com/jp/rc/>>

【発明の開示】

【発明が解決しようとする課題】

【0004】

しかしながら、個々のサービスは、それぞれの価値基準で専用の資源ノードの価値を決定し利用している。この場合、個々のサービス間で資源ノード価値基準が異なるため資源ノードの融通が難しいという問題があった。

【0005】

また、サービスの中でも資源ノードに優先度があり、優先度が低い資源ノードから他のサービスに渡して欲しいという要望があるが、トップダウン的な方式の場合、高い優先度で利用していた資源ノードが、他のサービスに割り当てられてしまう可能性があり、これまでおこなってきた処理が無駄になって再度やり直す必要が生じ、処理時間の長期化を招くという問題があった。

【0006】

この発明は、上述した従来技術による問題点を解消するため、効率的なブローカリングをおこなうことにより、個々のサービスの円滑な提供を実現することができる資源ブローカリングプログラム、該プログラムを記録した記録媒体、資源ブローカリング装置、および資源ブローカリング方法を提供することを目的とする。

【課題を解決するための手段】

【0007】

上述した課題を解決し、目的を達成するため、この発明にかかる資源ブローカリングプログラム、該プログラムを記録した記録媒体、資源ブローカリング装置、および資源ブローカリング方法は、各資源ノードに割り当てられた各サービスに応じた優先度を当該各資源ノードに対応付けて設定し、前記各資源ノードによる前記各サービスの処理時間の経過に応じて、当該各資源ノードについて設定された優先度を、より優先度の高いものに更新し、新たなサービスを割り当てる空きの資源ノードが存在しない場合に、サービスが既に割り当てられた第1の資源ノードと第2の資源ノードのうち、更新された優先度がより低い資源ノードに割り当てられているサービスを当該新たなサービスに切り替えることを特徴とする。

【0008】

また、上記発明において、当該各資源ノードについて設定された優先度を、当該各資源ノードの処理性能に応じて更新することとしてもよい。

【0009】

また、上記発明において、当該各資源ノードについて設定された優先度を、当該各資源

10

20

30

40

50

ノードの使用頻度に応じて大きくなるように更新することとしてもよい。

【0010】

また、上記発明において、当該各資源ノードについて設定された優先度を、当該各資源ノードの設置位置に応じて更新することとしてもよい。

【0011】

また、上記発明において、当該各資源ノードについて設定された優先度を、指定入力された実行時間内における当該各資源ノードの中断または終了の可否に応じて更新することとしてもよい。

【0012】

また、上記発明において、前記各サービスに関するジョブの集合のうち実行が完了したジョブの割合に応じて、当該各サービスが割り当てられた前記各資源ノードのうち前記ジョブの実行が完了していない資源ノードの優先度を、より優先度の高いものに更新することとしてもよい。

10

【0013】

また、上記発明において、当該各資源ノードについて設定された優先度を、当該各資源ノードの切替回数に応じて大きくなるように更新することとしてもよい。

【0014】

また、上記発明において、前記新たなサービスを割り当てる空きの資源ノードが存在しない場合に、サービスが既に割り当てられた複数の資源ノードのうち、更新された優先度が最低の資源ノードに割り当てられているサービスを当該新たなサービスに切り替えることとしてもよい。

20

【0015】

これらの発明によれば、個々のサービスの価値基準を統一的な価値基準に統一することができるため、サービスどうしの優先度を把握することができ、ブローカリングに反映させることができる。また、サービスに利用される資源ノードの価値がその状態に応じて変動するため、相対的に価値が低い資源ノードをサービス切替対象とし、価値が高い資源ノードをサービス切替対象から外すことができるため、個々のサービスの提供を効率的に実現することができる。

【発明の効果】

【0016】

本発明にかかる資源ブローカリングプログラム、該プログラムを記録した記録媒体、資源ブローカリング装置、および資源ブローカリング方法によれば、効率的なブローカリングをおこなうことにより、個々のサービスの円滑な提供を実現することができるという効果を奏する。

30

【発明を実施するための最良の形態】

【0017】

以下に添付図面を参照して、この発明にかかる資源ブローカリングプログラム、該プログラムを記録した記録媒体、資源ブローカリング装置、および資源ブローカリング方法の好適な実施の形態を詳細に説明する。

【0018】

(資源ブローカリングシステムのシステム構成図)

まず、この発明の実施の形態にかかる資源ブローカリングシステムのシステム構成について説明する。図1は、この発明の実施の形態にかかる資源ブローカリングシステムのシステム構成図である。図1において、資源ブローカリングシステム100は、資源ブローカリング装置101と、各サイトCに設置されている資源ノード102とが、ネットワーク110を介して通信可能となって接続されている。

40

【0019】

資源ブローカリング装置101は、複数のサービス間で利用される資源ノード102をブローカリングするコンピュータ装置である。具体的には、要求されるサービスに応じて、どのサイトCのどの資源ノード102を割り振るかを決定したり、あるサービスを提供

50

しているあるサイトCの資源ノード102を他のサービスに割り振ったりする。

【0020】

また、資源ノード102は各サイトCに設置されており、資源ブローカリング装置101によって割り与えられたサービスを端末(不図示)に提供するコンピュータ装置である。

【0021】

(コンピュータ装置のハードウェア構成)

つぎに、図1に示したコンピュータ装置のハードウェア構成について説明する。図2は、図1に示したコンピュータ装置のハードウェア構成を示すブロック図である。図2において、コンピュータ装置は、CPU201と、ROM202と、RAM203と、HDD(ハードディスクドライブ)204と、HD(ハードディスク)205と、FDD(フレキシブルディスクドライブ)206と、着脱可能な記録媒体の一例としてのFD(フレキシブルディスク)207と、ディスプレイ208と、I/F(インターフェース)209と、キーボード210と、マウス211と、スキャナ212と、プリンタ213と、を備えている。また、各構成部はバス200によってそれぞれ接続されている。

10

【0022】

ここで、CPU201は、コンピュータ装置の全体の制御を司る。ROM202は、ブートプログラムなどのプログラムを記憶している。RAM203は、CPU201のワークエリアとして使用される。HDD204は、CPU201の制御にしたがってHD205に対するデータのリード/ライトを制御する。HD205は、HDD204の制御で書き込まれたデータを記憶する。

20

【0023】

FDD206は、CPU201の制御にしたがってFD207に対するデータのリード/ライトを制御する。FD207は、FDD206の制御で書き込まれたデータを記憶したり、FD207に記憶されたデータをコンピュータ装置に読み取らせたりする。

【0024】

また、着脱可能な記録媒体として、FD207のほか、CD-ROM(CD-R、CD-RW)、MO、DVD(Digital Versatile Disk)、メモリカードなどであってもよい。ディスプレイ208は、カーソル、アイコンあるいはツールボックスをはじめ、文書、画像、機能情報などのデータを表示する。このディスプレイ208は、たとえば、CRT、TFT液晶ディスプレイ、プラズマディスプレイなどを採用することができる。

30

【0025】

I/F209は、通信回線を通じてインターネットなどのネットワーク110に接続され、このネットワーク110を介して他の装置に接続される。そして、I/F209は、ネットワーク110と内部のインターフェースを司り、外部装置からのデータの入出力を制御する。I/F209には、たとえばモデムやLANアダプタなどを採用することができる。

【0026】

キーボード210は、文字、数字、各種指示などの入力のためのキーを備え、データの入力をおこなう。また、タッチパネル式の入力パッドやテンキーなどであってもよい。マウス211は、カーソルの移動や範囲選択、あるいはウィンドウの移動やサイズの変更などをおこなう。ポインティングデバイスとして同様に機能を備えるものであれば、トラックボールやジョイスティックなどであってもよい。

40

【0027】

スキャナ212は、画像を光学的に読み取り、コンピュータ装置内に画像データを取り込む。なお、スキャナ212は、OCR機能を持たせてもよい。また、プリンタ213は、画像データや文書データを印刷する。プリンタ213には、たとえば、レーザプリンタやインクジェットプリンタを採用することができる。

【0028】

50

(資源ブローカリング装置101の機能的構成)

つぎに、この発明の実施の形態にかかる資源ブローカリング装置101の機能的構成について説明する。図3は、この発明の実施の形態にかかる資源ブローカリング装置101の機能的構成を示すブロック図である。図3において、資源ブローカリング装置101は、設定部301と、受付部302と、検出部303と、判断部304と、切替部305と、から構成されている。

【0029】

設定部301は、サービスごとに当該サービスに利用される資源ノード102の価値を所定の価値基準上において相対的に設定する。ここで、サービスとは、資源ノード102のコンピュータ端末に提供する情報処理であり、たとえば、データマイニングなどの非インタラクティブなサービスやインターネット電話やテレビ会議システムといったインタラクティブなサービスがある。

10

【0030】

また、資源ノード102の価値とは、利用するサービスにより資源ノード102に与えられる価値であり、サービスの優先度や重要度をあらわす指標である。個々のサービスでは資源ノード102の価値基準が異なるため、複数のサービスを同時に提供する場合には、あるサービスについて資源ノード102の不足が発生したとしても、どのサービスからの資源ノード102を融通すればよいか分からない。

【0031】

このため、設定部301では、個々のサービスに対して統一的な価値基準を用いて、各サービスで利用される価値をマッピングする。このマッピングは、要求されたサービス間で相対的に設定される。たとえば、インタラクティブなサービスと、非インタラクティブなサービスとの間では、インタラクティブなサービスに利用される資源ノード102の価値が、非インタラクティブなサービスに利用される資源ノード102の価値よりも大きくなるように設定される。

20

【0032】

また、設定部301は、資源ノード102の価値が、当該価値が与えられている資源ノード102の状態に基づいて変動するように設定することもできる。たとえば、あるサービスについてジョブを実行している資源ノード102について、ジョブの実行時間が長いほど、途中で他のサービスに切り替えられると、これまでのジョブが無駄となる。したがって、資源ノード102の価値を実行時間に応じて大きくなるように変動させることで、当該資源ノード102を切替対象外とすることが可能となる。

30

【0033】

同様に、処理性能(CPUのクロック周波数やメモリ容量)が高い資源ノード102ほど価値が大きくなるように設定してもよく、また、使用頻度が高い資源ノード102ほど価値が大きくなるように設定してもよい。また、遠隔地に設置されている資源ノード102については、物理的に通信距離が長くなるため、他の資源ノード102と相対的に遠いサイトCに設置されている資源ノード102については、価値が低くなるように設定することで、サービス提供の高速化を実現することができる。

【0034】

40

また、設定部301は、資源ノード102の価値が、指定入力された資源ノード102の実行時間に応じて変動するように設定することもできる。実行時間の指定入力は、ユーザによってサービスに対するジョブ投入時などにおこなわれる。このとき指定入力される実行時間は、サービスをおこなう具体的な実行時間であってもよいし、サービスをおこなう時間帯であってもよい。

【0035】

実行時間が指定入力されている資源ノード102については、当該実行時間内において、他の資源ノード102の価値よりも高く価値を設定する。さらに、実行時間が指定入力されている場合において、実行時間内における資源ノード102の中断または終了の可否に応じて変動するように、価値を設定するようにしてもよい。

50

【 0 0 3 6 】

すなわち、実行時間内であっても、資源ノード102の中断または終了に関する指定入力があった場合には、価値を低く設定することができる。このとき、資源ノード102の価値を漸次的に低下させるようにしてもよいし、資源ノード102の価値として取り得る範囲のうち最も低い値まで低下させるようにしてもよい。

【 0 0 3 7 】

また、設定部301は、他のサービスに関するジョブの集合のうち実行が完了したジョブの割合に応じて、他のサービスに利用される資源ノード102のうちジョブの実行が完了していない資源ノード102の価値が大きくなるように設定することもできる。すなわち、他のサービスについてジョブの集合が投入された場合、当該ジョブの集合のうち実行が完了したジョブの割合に応じて、ジョブを実行中の資源ノード102の価値が大きくなるように設定する。

10

【 0 0 3 8 】

さらに、設定部301は、資源ノード102の切替回数に応じて、資源ノード102の価値が大きくなるように設定するようにしてもよい。具体的には、あるサービスに利用されているにもかかわらず、異なるサービスに利用される資源ノード102に切り替えられた回数に応じて、資源ノード102の価値が大きくなるように設定する。

【 0 0 3 9 】

このように、資源ノード102の切替回数の増加にともなって資源ノード102の価値を大きくすることにより、切り替えによるジョブの無駄を低減し、ジョブの実行完了にかかる時間を短縮することができる。

20

【 0 0 4 0 】

また、受付部302は、複数のサービスのうちのサービスの要求を受け付ける。具体的には、サービスの種類、利用したい資源ノード102の数などを要求として受け付ける。この要求は、資源ブローカリング装置101に直接入力することで、受け付けてもよく、また、ネットワーク110を介して外部のコンピュータ装置から受信することとしてもよい。

【 0 0 4 1 】

また、検出部303は、受付部302によって要求が受け付けられた一のサービスに利用される資源ノード102が不足しているか否かを検出する。初期においては、資源ノード102の割り当ては、要求の順番や各サービスの価値により順次決定されるが、要求されているサービスに利用したい資源ノード102の数が、資源ノード102の総数を超えている場合がある。また、初期では資源ノード102が充足していても、個々のサービスを運用していく過程で要求されているサービスに利用したい資源ノード102の数が、資源ノード102の総数を超えている場合がある。このような場合に資源ノード不足を検出することとなる。

30

【 0 0 4 2 】

また、判断部304は、検出部303によって不足していることが検出された場合、設定部301によって設定された一のサービスに利用されている資源ノード102の価値が、設定部301によって設定された一のサービス以外の他のサービスに利用されている資源ノード102の価値よりも大きいか否かを判断する。

40

【 0 0 4 3 】

資源ノード不足となった場合、価値が高いサービスに利用される資源ノード102が要求されていると、それよりも価値の低いサービスに利用されている資源ノード102を割り当てる必要がある。そのため、価値の大小を判断することで、サービスの切替対象となる資源ノード102を特定することができる。

【 0 0 4 4 】

また、切替部305は、判断部304によって判断された判断結果に基づいて、資源ノード102の切替をおこなう。すなわち、要求している一のサービスに利用される資源ノード102が不足している場合、その資源ノード102の価値よりも低い価値となる他の

50

サービスに利用されている資源ノード102を、一のサービスに利用される資源ノード102に切り替える。また、他のサービスに利用されている資源ノード102が複数存在する場合、その中で最も価値が低い資源ノード102を、一のサービスに利用される資源ノード102に切り替えるのが好ましい。

【0045】

(資源ブローカリング処理手順)

つぎに、この発明の実施の形態にかかる資源ブローカリング装置101の資源ブローカリング処理手順について説明する。図4は、この発明の実施の形態にかかる資源ブローカリング装置101の資源ブローカリング処理手順を示すフローチャートである。

【0046】

図4において、まず、設定部301により各サービスの価値を設定し(ステップS401)、受付部302により一のサービスの要求があるまで待ち受ける(ステップS402: No)。そして、一のサービスの要求があった場合(ステップS402: Yes)、検出部303により資源ノード不足か否かを検出する(ステップS403)。資源ノード不足が検出されない場合(ステップS403: No)、ステップS406に移行する。

【0047】

一方、資源ノード不足が検出された場合(ステップS403: Yes)、判断部304により、一のサービスに利用される資源ノード102の価値(V1)が他のサービスにおいて現在利用中の資源ノード102の価値(V2)より高いか否かを判断する(ステップS404)。V1 > V2でない場合(ステップS404: No)、ステップS406に移行する。一方、V1 > V2である場合(ステップS404: Yes)、現在利用中の資源ノード102が提供する他のサービスを一のサービスに切り替える(ステップS405)。

【0048】

このあと、一のサービスが終了したか否かを判断し(ステップS406)、終了していない場合には(ステップS406: No)、ステップS402に戻る。一方、一のサービスが終了した場合には(ステップS406: Yes)、一連の処理を終了する。

【0049】

(資源ブローカリング処理の具体例)

つぎに、資源ブローカリング処理の具体例について図5～図10を用いて説明する。図5～図10は、資源ブローカリング処理の具体例を示す説明図である。ここでは一例として、サービスAとサービスBを提供することとする。サービスAは、非インタラクティブなデータマイニングであり、サービスBは、インタラクティブなインターネット電話とする。

【0050】

また、サービスAに利用される資源ノード102の価値Vaは、 $1 \leq V_a \leq 70$ とする。価値VaはサービスAのジョブ実行時間の長さに応じて増加する。なお、価値Vaの初期値はVa = 1とする。

【0051】

また、サービスBの価値Vbは、以下の通り、2値の値をとる。

Vb = 100 (資源ノード実行)

Vb = 0 (資源ノード実行終了)

【0052】

図5～図10において、サービスAの価値基準Xaは価値Vaの範囲を特定しており、サービスBの価値基準Xbは価値Vbの範囲を特定している。価値基準Xaと価値基準Xbが共通の価値基準Xにマッピングされている。また、丸図形は資源ノード102をあらわしており、丸図形内の数字は価値をあらわしている。

【0053】

特に、数字が1以上70以下である丸図形はサービスAに利用されている資源ノード102をあらわしており、数字が100である丸図形はサービスAに利用されている資源ノ

10

20

30

40

50

ード102をあらわしている。また、価値基準 X_a 、 X_b 上の点線の丸図形は、要求されている資源ノード102、すなわち不足分の資源ノード102をあらわしている。

【0054】

まず、図5では、数字が「1」である点線の丸図形が4つあるため、サービスAでは資源ノード102が4台不足している。また、図5の段階においては、サービスAに利用されている資源ノード102の価値 V_a は $V_a = 10, 20, 30, 45, 70$ に変化している。すなわち、 $V_a = 70$ によってあらわされる資源ノード102が最も長時間データマイニングの実行を継続していることをあらわしている。なお、図5においては、サービスBは資源ノード102を要求していないものとする。

【0055】

図6において、 $V_a = 70$ によってあらわされる資源ノード102によるデータマイニングの実行が終了したとすると、その資源ノード102の価値 V_a は70から1に戻る。また、これにともない、価値 $V_a = 45$ の資源ノード102の価値 V_a は $V_a = 60$ に引き上げられる。

【0056】

図7においては、サービスBに利用される資源ノード102が2台要求されている。サービスBに利用される資源ノード102の価値 V_b は $V_b = 100$ であるので、 $V_b > V_a$ となり、サービスAよりも優先して実行されることとなる。

【0057】

このため、図8においては、サービスAに利用されている資源ノード102のうち価値 V_a が小さい資源ノード102がサービスBに割り当てられる。ここでは、サービスBに利用したい資源ノード102は2つであるため、サービスAに利用されている8つの資源ノード102 ($V_a = 1, 10, 10, 20, 20, 30, 30, 60$)のうち、価値 V_a が小さい資源ノード102 ($V_a = 1, 10$)がサービスBに割り振られることとなる。すなわち、サービスAに利用されている資源ノード102のうち、ジョブの実行時間が長い資源ノード102ほどサービスBへの切替対象外となる。

【0058】

また、図9において、サービスBに利用されている2台の資源ノード102のうち1台の資源ノード102が実行終了した場合、その資源ノード102の価値 V_b は100から0に引き下がる。そして、図10において、サービスBの資源ノード102の要求がなく、サービスAの資源ノード102の要求がある場合、0になったサービスBの資源ノード102をサービスAに割り当てる。

【0059】

この図5～図10までのブローカリング処理により、サービスA、Bを同時期に提供する場合において、資源ノード102の有効活用が実現でき、特に、サービスAではデータマイニングのジョブ実行時間の短縮化を図ることができ、サービスBでは、受付可能なインターネット電話の数の増加を図ることができる。

【0060】

つぎに、上述した資源ブローカリング処理の適用前後におけるサービス処理状態について説明する。図11-1～図11-4は、資源ブローカリング処理の適用前後におけるサービス処理状態を示すグラフである。図11-1は、サービスAにおける資源ブローカリング処理の適用前のサービス処理状態を示すグラフであり、図11-2は、サービスAにおける資源ブローカリング処理の適用後のサービス処理状態を示すグラフである。各グラフGA1、GA2において、横軸は時間、縦軸はジョブ実行数をあらわしている。

【0061】

グラフGA1、GA2において、黒塗り部分は待機中を示しており、網掛け部分はジョブ実行中を示している。グラフGA1、GA2を比較すると、グラフGA2の方が黒塗り部分の領域が少ないため、効率的にデータマイニングが実行されていることがわかる。

【0062】

また、図11-3は、サービスBにおける資源ブローカリング処理の適用前のサービス

10

20

30

40

50

処理状態を示すグラフであり、図 1 1 - 4 は、サービス B における資源ブローカリング処理の適用後のサービス処理状態を示すグラフである。各グラフ G B 1 , G B 2 において、横軸は時間、縦軸は接続要求数をあらわしている。

【 0 0 6 3 】

グラフ G B 1 , G B 2 において、黒塗り部分は受付拒否を示しており、ハッチング部分は受付（ジョブ実行中）を示している。グラフ G B 1 , G B 2 を比較すると、グラフ G B 2 の方が黒塗り部分の領域が少ないため、効率的にインターネット電話によるインタラクティブ通信が実行されていることがわかる。

【 0 0 6 4 】

つぎに、上述した資源ブローカリング処理の適用前後における資源ノード 1 0 2 の割り当て状態について説明する。図 1 2 - 1 および図 1 2 - 2 は、資源ブローカリング処理の適用前後における資源ノード 1 0 2 の割り当て状態を示すグラフである。図 1 2 - 1 において、グラフ G C 1 は、資源ブローカリング処理の適用前のグラフであり、図 1 2 - 2 において、グラフ G C 2 は、資源ブローカリング処理の適用後のグラフである。なお、両グラフ G C 1 , G C 2 において、横軸は時間、縦軸は割り当てられた資源ノード 1 0 2 の数をあらわしている。

10

【 0 0 6 5 】

両グラフ G C 1 , G C 2 において、黒塗り部分はサービス A をあらわしており、網掛け部分はサービス B をあらわしている。適用前のグラフ G C 1 では、サービスごとに資源ノード数が固定されているため資源ノード数に時間的变化はないが、適用後のグラフ G C 2 20
では、サービス B をサービス A よりも優先して割り当てるため、サービス B に利用される資源ノード数が増加すると、サービス A に利用される資源ノード数が減少する。

【 0 0 6 6 】

このように、この発明の実施の形態によれば、個々のサービスの価値基準を統一的な価値基準に統一することができるため、サービスどうしの優先度を把握することができ、ブローカリングに反映させることができる。また、サービスに利用される資源ノード 1 0 2 の価値がその状態に応じて変動するため、相対的に価値が低い資源ノード 1 0 2 をサービス切替対象とし、価値が高い資源ノード 1 0 2 をサービス切替対象から外すことができるため、サービス間で資源ノード 1 0 2 を効率的に融通しあうことができる。これにより、個々のサービスの円滑な提供を実現することができる。

30

【実施例 1】

【 0 0 6 7 】

つぎに、資源ブローカリングシステム 1 0 0 の実施例 1 について説明する。図 1 3 は、資源ブローカリングシステム 1 0 0 の詳細なシステム構成図である。資源ブローカリング装置 1 0 1 は、グリッド・サービスサブシステム 1 3 0 1 と、資源ブローカリングサブシステム 1 3 0 2 と、グリッド情報サブシステム 1 3 0 3 と、運用管理サブシステム 1 3 0 4 と、から構成されている。

【 0 0 6 8 】

グリッド・サービスサブシステム 1 3 0 1 とは、グリッドで実行する個々のサービスを実現するサブシステムであり、サービスの種類ごとに 1 つずつ用意される。グリッドとは、地理的に分散した複数のコンピュータシステムをネットワークで結び、計算パワーを提供する仮想的な 1 つのシステムとして見せる技術である。グリッド・サービスサブシステム 1 3 0 1 は、既存のアプリケーションをグリッド環境に対応させ、資源ブローカリング装置 1 0 1 上のサービスとして実行できるようにする。

40

【 0 0 6 9 】

また、資源ブローカリングサブシステム 1 3 0 2 とは、グリッド・サービスサブシステム 1 3 0 1 からの資源要求を受け付けて、サービスの実行に必要な物理的な資源ノード 1 0 2 をブローカリングする。優先度が高いサービスの資源要求が満たされるように、各サービスへの資源割り当ての配分を動的に調整する。また、資源要求をサービスや資源の優先度に基づいて調停する機能や、実行するアプリケーションを個々の資源ノード 1 0 2 ごとに

50

とに切り替える機能を備えている。

【 0 0 7 0 】

また、グリッド情報サブシステム 1 3 0 3 とは、資源ブローカリング装置 1 0 1 内の様々な情報を収集して提供するサブシステムである。たとえば、個々の資源ノード 1 0 2 に関する情報（CPU の性能や OS の種類）や各サービスに関する情報（負荷や資源ノード 1 0 2 の獲得状況）の収集と提供をおこなう。

【 0 0 7 1 】

また、運用管理サブシステム 1 3 0 4 とは、資源ブローカリング装置 1 0 1 の運用管理をおこなうためのサブシステムである。資源ブローカリング装置 1 0 1 の全体的な動作状況の確認や運用ポリシーの設定をおこなう。

10

【 0 0 7 2 】

つぎに、資源ブローカリングシステム 1 0 0 を構成するモジュールについて説明する。図 1 3 中、ライフサイクルマネージャ LM #（# は 1 ~ N までの番号）は、グリッド・サービスサブシステム 1 3 0 1 のモジュールである。ライフサイクルマネージャ LM # は、個々のサービスごとに 1 つ用意されており、サービスの開始から終了まで、サービスに割り当てられた資源ノード 1 0 2 を管理する。また、サービスの負荷の変動に応じて、資源ノード 1 0 2 の追加や解放をアービトラータ ARB に要求する。また、管理している資源ノード 1 0 2 の優先度を自律的に調節する機能も有する。

【 0 0 7 3 】

また、ライフサイクルマネージャファクトリサービス LMFS は、資源ブローカリングサブシステム 1 3 0 2 のモジュールであり、サービスの起動と停止をおこなう。サービスの起動要求を受け付けると、そのサービスのライフサイクルマネージャ LM # を実行するための資源ノード 1 0 2 を要求し、割り当てられた資源ノード 1 0 2 でライフサイクルマネージャ LM # を起動する。また、サービスの停止要求を受け付けると、そのサービスのライフサイクルマネージャ LM # を停止して資源ノード 1 0 2 を解放する。

20

【 0 0 7 4 】

また、アービトラータ ARB は、資源ブローカリングサブシステム 1 3 0 2 のモジュールであり、ライフサイクルマネージャ LM # からの資源ノード 1 0 2 の追加 / 解放要求を受け付けて、各サービスに対して資源ノード 1 0 2 を割り当てる。また、各サービスの優先度に基づいて調停をおこない、優先度の高いサービスにグリッドの計算パワーを集中させる。

30

【 0 0 7 5 】

また、物理資源ブローカ PRB は、資源ブローカリングサブシステム 1 3 0 2 のモジュールである。物理資源ブローカ PRB は、グリッド内の各資源ノード 1 0 2 の物理的な属性情報に基づいて、サービスを実行する能力や機能を持った資源ノード 1 0 2 をアービトラータ ARB にブローカリングする。

【 0 0 7 6 】

また、資源ロールスイッチャ RRS は、資源ブローカリングサブシステム 1 3 0 2 のモジュールである。資源ロールスイッチャ RRS は各資源ノード 1 0 2 が実行するサービス（アプリケーション）の切り替えを実行する。

40

【 0 0 7 7 】

また、ノードモニタ NM は、グリッド情報サブシステム 1 3 0 3 のためのモジュールである。ノードモニタ NM は、各資源ノード 1 0 2 に 1 つずつ配置され、資源ノード 1 0 2 の情報（CPU の種類や負荷、メモリ使用率など）を収集し、定期的にクラスタマネージャ CM に報告する。また、物理切替機 ASCC は資源ブローカリングサブシステム 1 3 0 2 における論理的な切替処理に応じて、物理的に資源ノード 1 0 2 のサービス切替処理をおこなう。

【 0 0 7 8 】

また、クラスタマネージャ CM は、グリッド情報サブシステム 1 3 0 3 のためのモジュールであり、各サイト C に 1 つずつ配置される。クラスタマネージャ CM は、サイト C 内

50

のノードモニタNMから集まる情報をルートサーバRSに中継する。

【0079】

また、ルートサーバRSは、グリッド情報サブシステム1303のモジュールであり、グリッド内のすべての資源ノード102の情報を集約する。また、アーカイバARは、グリッド情報サブシステム1303のモジュールであり、ルートサーバRSに集約された情報を蓄積し、データベース化するモジュールである。資源ブローカリングサブシステム1302に対して、データベースの検索機能を提供する。

【0080】

また、アプリケーションラッパーAWは、資源ブローカリングサブシステム1302のためのモジュールであり、グリッドの各資源ノード102に配置するモジュールで、資源ノード102が実行するアプリケーションのAPIをラッピングし、アプリケーションとライフサイクルマネージャLM#を接続する。

10

【0081】

また、アドミニストレーションポータルAPT Lは、資源ブローカリングサブシステム1302のモジュールであり、グリッドで実行するサービスの管理者がサービスの起動や停止をおこなうためのインターフェースを提供する。

【0082】

また、アドミニストレーションコンソールACNSは、運用管理サブシステム1304のモジュールであり、資源ブローカリング装置101の管理者が資源ブローカリングシステム100全体の設定や調整をおこなうためのインターフェースを提供する。

20

【0083】

つぎに、実施例1における資源ブローカリング処理について説明する。図14は、実施例1における資源ブローカリング処理を示すシーケンス図である。図14に示した例では、資源ノード102の要求と割り当てに関する典型的な動作シーケンスを示す。この例では、サービスsの優先度がサービスtの優先度よりも高いと仮定している。なお、カッコ数字はシーケンスの順序をあらわしている。

【0084】

アービトレータARBは各ライフサイクルマネージャLM#からの資源ノード要求をサービスの優先度に基づいて調停するため、サービスsのライフサイクルマネージャLM#(以下、「LMs」と表記する)の要求がサービスtのライフサイクルマネージャLM#(以下、「LMt」と表記する)の要求よりも優先的に扱われる。

30

【0085】

図14では、アービトレータARBでの調停の結果、サービスsに割り当てられている資源ノード102をサービスtの資源ノード102に切り替えることが決定され、物理資源ブローカPRB、資源ロールスイッチャRRS、物理切替機ASCC、アプリケーションラッパーAWの各モジュールが連携して切替をおこなうシーケンスをあらわしている。

【0086】

この実施例1によれば、トータルコストの削減を図ることができる。具体的には、別々のシステムとして構築されているサーバを1つのシステムに統合できたり、地理的に分散して設置されているサーバを1つのシステムとして統合できたり、既存のアプリケーションを容易にグリッド環境へ移行することができたり、複数のサービス間で資源ノード102の余裕を融通しあうことによって、各々のサービスのピーク性能を向上させることができる。また、1つの共通の手法で複数のサービスを管理することができる。

40

【0087】

また、ビジネスの状況の変化に柔軟に対応できるシステムを実現することができる。具体的には、必要量に応じた計算パワーを自動的にサービスに供給することができたり、優先度の高いサービスに計算パワーを自動的に集中することができたり、状況の変化に合わせて自律的にサービスの優先度を調整することができる。

【実施例2】

【0088】

50

つぎに、資源ブローカリングシステム100の実施例2について説明する。実施例2では、資源ブローカリングシステム100のアービトラータARB(図13参照)において設定される資源ノード102の優先度について具体的に説明する。

【0089】

サービスに利用される資源ノード102の優先度は、ジョブの実行時間、資源ノード102の処理性能および他の資源ノード102との相対的な通信距離などに応じて設定される。実施例2においては、さらに他の価値基準を追加して資源ノード102の優先度を設定する。

【0090】

(優先度設定の具体例(その1))

まず、サービスに対するジョブ投入時に指定入力されたジョブの実行時間に応じて資源ノード102の優先度を設定する場合について説明する。ここで、ジョブの実行時間とは、ジョブの実行に費やす具体的な時間(たとえば、2時間)であってもよいし、ジョブを実行する時間帯(たとえば、15時~17時)であってもよい。

【0091】

ジョブの実行時間は、ジョブ投入時にサービスの管理者によってグリッド・サービスサブシステム1301に指定入力される。資源ブローカリングサブシステム1302のアービトラータARBは、グリッド・サービスサブシステム1301のライフサイクルマネージャLM#から、指定入力されたジョブの実行時間を受け付け、資源ノード102の優先度を設定する。

【0092】

このとき、ジョブの実行時間が指定された資源ノード102については、指定されたジョブの実行時間内において、他の資源ノード102の優先度よりも高く優先度を設定する。すなわち、ジョブの実行時間が指定された資源ノード102の優先度は、指定されたジョブの実行時間内においては、他の資源ノード102よりも価値が大きく増加することとなる。

【0093】

なお、価値の上限値は、個々のサービスに利用される資源ノード102の価値の上限値とする。たとえば、サービスAに利用される資源ノード102の価値 V_a が $1 \leq V_a \leq 70$ であった場合、上限値は $V_a = 70$ となる。

【0094】

また、ジョブの実行時間が経過した場合には、すぐに資源ノード102を解放するのではなく、漸次的に優先度を低下させていく。すなわち、ジョブの実行時間が経過した場合、資源ノードの価値を下限値またはサービスが要求する価値よりも低い値に設定するのではなく、緩やかに価値を低下させていく。

【0095】

ここで、指定されたジョブの実行時間に応じて優先度を設定する場合の資源ブローカリング処理の具体例について図15~図18を用いて説明する。図15~図18は、実施例2における資源ブローカリング処理の具体例を示す説明図である。

【0096】

ここでは一例として、サービスAとサービスBを提供することとする。また、サービスAに利用される資源ノード102の価値 V_a は、 $1 \leq V_a \leq 70$ とする。また、サービスBの価値 V_b は、 $V_b = 100$ (資源ノード実行)または $V_b = 0$ (資源ノード実行終了)の2値の値をとる。

【0097】

また、図15~図18において、サービスAの価値基準 X_a は価値 V_a の範囲を特定しており、サービスBの価値基準 X_b は価値 V_b の範囲を特定している。価値基準 X_a と価値基準 X_b が共通の価値基準 X にマッピングされている。

【0098】

図15において、サービスAについてジョブを実行している資源ノード102が4台あ

10

20

30

40

50

り、各資源ノード102の価値 V_a は $V_a = 10$ である。価値 V_a は、サービスAのジョブ実行時間の長さに応じて増加するとともに、ジョブの実行時間が指定されている場合にはそれにともなう値が付加される。

【0099】

具体的には、サービスAについてジョブを実行している4台の資源ノード102のうち、2台の資源ノード102（実行時間指定ジョブ1501）についてジョブの実行時間が指定されている。なお、ジョブの実行時間が指定されていない2台の資源ノード102を実行時間指定無しジョブ1502とする。

【0100】

このため、実行時間指定ジョブ1501については、ジョブ実行時間の長さに応じて価値 V_a が増加するとともに、指定されたジョブの実行時間中は、それにともなう値が付加される。また、実行時間指定無しジョブ1502については、ジョブの実行時間が指定されていないため、実際に経過したジョブ実行時間の長さに応じてのみ価値 V_a が増加する。

10

【0101】

図16において、一定時間経過後は、実行時間指定無しジョブ1502について、ジョブ実行時間の経過にともない価値 V_a が $V_a = 10$ から $V_a = 20$ に引き上げられている。また、実行時間指定ジョブ1501については、ジョブ実行時間の経過にともない価値 V_a が $V_a = 10$ から $V_a = 20$ に引き上げられるとともに、ジョブの実行時間の指定にともない、さらに値が付加されて $V_a = 10$ から $V_a = 30$ に引き上げられている。

20

【0102】

図17において、さらに時間が経過して（指定されたジョブの実行時間 - t ）となった場合には、実行時間指定ジョブ1501について、価値 V_a が上限値である $V_a = 70$ に引き上げられており、これ以上は増加しない。また、実行時間指定無しジョブ1502については、ジョブ実行時間の経過にともない価値 V_a が $V_a = 20$ から $V_a = 50$ に引き上げられている。

【0103】

図18において、ジョブの実行時間経過後は、実行時間指定ジョブ1501について、指定されたジョブの実行時間が経過した場合には、すぐに資源ノード102（実行時間指定ジョブ1501）を解放するのではなく、漸次的に価値 V_a を低下させていく。ここでは、実行時間指定ジョブ1501の価値 V_a は、 $V_a = 70$ から $V_a = 20$ に低下している。また、実行時間指定無しジョブ1502については、ジョブ実行時間の経過にともない価値 V_a が $V_a = 50$ から $V_a = 60$ に引き上げられている。

30

【0104】

この図15～図18のプロローキング処理によれば、指定されたジョブの実行時間中は、資源ノード102の優先度（価値）をより高く設定することができる。このため、実行時間が短いために優先度が低いジョブであっても、指定した実行時間内（特に、ジョブ実行の完了が近づけば近づくほど）においては、より確実に最後までジョブを実行させることができる。

【0105】

すなわち、実行時間の経過にともなう優先度の増加が少ないジョブについて、実行途中に他のサービスに切り替えられることによるジョブ実行の再試行を防ぐことができる。これにより、各資源ノード102における無駄な計算処理を低減させることができる。

40

【0106】

（優先度設定の具体例（その2））

つぎに、外界との相互作用を有するジョブの実行中に、ジョブ実行の中断または終了の可否を指定することができるGUI（Graphical User Interface）をユーザに提供する場合について説明する。

【0107】

具体的には、グリッド・サービスサブシステム1301のアプリケーションを利用して

50

、サービス用のコンピュータ装置に上記G U Iを表示させることにより、サービスの管理者に資源ノード102の解放に関する許可を得る。すなわち、アービトラータA R Bにおいて資源ノード102の解放を自動的におこなうのではなく、サービスの管理者による解放の許可に基づいて、資源ノード102の解放を判断する。

【0108】

まず、外界との相互作用を有するテレビ会議を実行する場合における優先度の設定について説明する。図19は、テレビ会議システムにおける優先度設定の概要を示す説明図である。ここでは、サービスの管理者によってテレビ会議の実行時間(終了時間)が予め指定されていることとする。

【0109】

まず、アービトラータA R Bは、指定されたテレビ会議の終了時間の5分前となった場合、グリッド・サービスサブシステム1301のアプリケーションに対して、ジョブ実行の中断の可否を指定することができるG U Iの表示要求をおこなう(1)。なお、G U Iの表示要求をおこなう時間は任意に設定可能であり、たとえば、テレビ会議の終了時間の10分前であってもよい。

【0110】

つぎに、表示要求を受け付けたグリッド・サービスサブシステム1301のアプリケーションは、テレビ会議用のコンピュータ装置のディスプレイにG U Iを表示させる(2)。

【0111】

ここで、テレビ会議の参加者によるテレビ会議の終了に関する指定入力を受け付け(3)、受け付けた指定入力をアービトラータA R Bに送信する(4)。そして、アービトラータA R Bは、受け付けた指定入力に基づいて、資源ノード102の優先度を設定する(5)。

【0112】

ここで、グリッド・サービスサブシステム1301のアプリケーションを利用してサービス用のコンピュータ装置に表示されるG U Iについて説明する。図20は、コンピュータ装置に表示されるG U Iの一例を示す説明図である。

【0113】

図20において、G U I上には、テレビ会議の終了予定時刻が表示されており、この時間通りにテレビ会議を終了するか否かを指定する選択肢が表示されている。なお、ここで表示されているテレビ会議の終了予定時刻は、予め指定されているテレビ会議の実行時間である。

【0114】

ここで、テレビ会議を予定通り終了する場合には、G U I上の「はい」を指定する。具体的には、たとえば、テレビ会議の参加者によってG U I上に表示されているカーソルを操作して、「はい」に対応するアイコンをクリックする。

【0115】

この場合、アービトラータA R Bは、予め指定されているテレビ会議の実行時間に基づいて、テレビ会議を終了させて、資源ノード102を解放する。ここでは、アービトラータA R Bは、5分後にテレビ会議を終了させて資源ノード102を解放することとなる。

【0116】

また、予定を変更してテレビ会議を延長する場合には、G U I上の「いいえ」を指定して、申請する延長時間を入力する。この場合、アービトラータA R Bは、入力された延長時間に基づいて、資源ノード102の優先度を設定する。

【0117】

具体的には、延長時間が入力された場合には、延長時間が経過するまでの間は、資源ノード102の優先度を高く設定し、他のサービスへ割り当てられることを防ぐ。なお、延長時間が経過した場合には、アービトラータA R Bは、テレビ会議を終了させて資源ノード102を解放する。

10

20

30

40

50

【0118】

つぎに、外界との相互作用を有するバッチ処理を実行する場合における優先度の設定について説明する。図21は、バッチ処理システムにおける優先度設定の概要を示す説明図である。ここでは、サービスの管理者によってバッチ処理の実行時間（終了時間）が予め指定されていることとする。

【0119】

また、バッチ処理システム（複数のジョブの集合）において、各ジョブが占める実行割合が明らかになっていることとする。さらには、バッチ処理を中断または終了させるか否かを問い合わせるインターフェースを有している。

【0120】

まず、アービトレータARBは、指定されたバッチ処理の終了時間の1分前となった場合、グリッド・サービスサブシステム1301のアプリケーションに対してバッチ処理を中断または終了させるか否かを問い合わせる（1）。

【0121】

つぎに、問い合わせを受け付けたグリッド・サービスサブシステム1301のアプリケーションは、バッチ処理を実行中のジョブ実行計算機2101に対して、バッチ処理を中断または終了させるか否かを問い合わせる（2）。

【0122】

問い合わせを受け付けたジョブ実行計算機2101は、バッチ処理において実行する計算のうち90%が終了しているか否かを判断する（3）。そして、その判断結果を問い合わせ結果として、グリッド・サービスサブシステム1301のアプリケーションに返信する（4）。すなわち、バッチ処理を中断可能であるか否かを示す問い合わせ結果を返信する。

【0123】

つぎに、グリッド・サービスサブシステム1301のアプリケーションは、受け付けたジョブ実行計算機2101からの問い合わせ結果を、アービトレータARBに送信する（5）。そして、アービトレータARBは、受け付けた問い合わせ結果に基づいて、資源ノード102の優先度を設定する（6）。

【0124】

具体的には、アービトレータARBは、バッチ処理を中断することができないことを示す問い合わせ結果を受け付けた場合には、資源ノード102の優先度を高く設定することによって他のサービスへ割り当てられることを防ぐ。

【0125】

また、アービトレータARBは、バッチ処理を中断することができることを示す問い合わせ結果を受け付けた場合には、バッチ処理を終了させて、資源ノード102を解放する。ここでは、アービトレータARBは、1分後にバッチ処理を終了させて資源ノード102を解放することとなる。

【0126】

このように、流動的に変化するサービスの状況に応じて資源ノード102の優先度を設定することによって、より効率的なブローカリングをおこなうことができる。たとえば、ジョブの実行時間について延長する指定入力があった場合などには、資源ノード102の優先度を高くすることによって、他のサービスに切り替えられることを防ぐことができる。

【0127】

（優先度設定の具体例（その3））

つぎに、サービスについてジョブの集合が投入された場合に、当該ジョブの集合のうち実行が完了したジョブの割合に応じて優先度を設定する場合について説明する。ジョブの集合とは、サービスを提供するために複数のジョブの実行を必要とする場合に投入されるジョブ群である。たとえば、ある空間を探索する際における各部分空間の探索ジョブの集まりをジョブの集合という。

10

20

30

40

50

【 0 1 2 8 】

ジョブの集合のうち少数のジョブの実行が終了しないために、全体ジョブ（集合全体）の実行にかかる時間が増大するという問題が発生する。ここでは、ジョブの集合のうち実行中のジョブについての優先度を、実行が完了したジョブの割合に応じて高く設定し、全体ジョブの実行が早期に終了するようにする。

【 0 1 2 9 】

具体的には、アービトラータ A R B は、グリッド・サービスサブシステム 1 3 0 1 のライフサイクルマネージャ L M # から、ジョブの集合を区別する識別子および各ジョブの実行状況を受け付ける。そして、アービトラータ A R B は、受け付けたジョブの集合を区別する識別子および各ジョブの実行状況に基づいて、現在実行中のジョブについての優先度を設定する。

10

【 0 1 3 0 】

ここで、ジョブの集合のうち実行が完了したジョブの割合に応じて優先度を設定する場合の資源ブローカリング処理の具体例について図 2 2 ~ 図 2 5 を用いて説明する。図 2 2 ~ 図 2 5 は、実施例 2 における資源ブローカリング処理の他の具体例を示す説明図である。

【 0 1 3 1 】

図 2 2 において、サービス A について関連するジョブが 8 つ（ジョブの集合）投入され、そのうちの 2 つのジョブが実行されている。すなわち、投入された 8 つのジョブのうち、2 つのジョブに資源ノード 1 0 2 が割り当てられている。このとき、ジョブに割り当てられている 2 台の資源ノード 1 0 2 の価値 V_a は $V_a = 1$ である。

20

【 0 1 3 2 】

図 2 3 において、一定時間経過後、投入された 8 つのジョブのうち 4 つのジョブに資源ノード 1 0 2 が割り当てられた結果、そのうち 2 つのジョブの実行が終了している。これにより、ジョブを実行中の 2 台の資源ノード 1 0 2 の価値 V_a が $V_a = 1$ から $V_a = 2$ に引き上げられている。

【 0 1 3 3 】

すなわち、投入されたジョブのうち実行が完了したジョブの割合（8分の2）が大きくなったため、ジョブの実行が完了していない資源ノード 1 0 2 の価値 V_a が引き上げられている。なお、ジョブ実行時間の経過にともなう価値の増加も含まれている（以下、同じ）。

30

【 0 1 3 4 】

図 2 4 において、さらに一定時間経過後、投入された 8 つのジョブのうち 6 つのジョブに資源ノード 1 0 2 が割り当てられた結果、そのうち 4 つのジョブの実行が終了している。これにより、ジョブを実行中の 2 台の資源ノード 1 0 2 の価値 V_a が $V_a = 2$ から $V_a = 4$ に引き上げられている。

【 0 1 3 5 】

すなわち、投入されたジョブのうち実行が完了したジョブの割合（8分の4）が大きくなったため、ジョブの実行が完了していない資源ノード 1 0 2 の価値 V_a が引き上げられている。

40

【 0 1 3 6 】

図 2 5 において、さらに一定時間経過後、投入された 8 つのジョブのうち 7 つのジョブに資源ノード 1 0 2 が割り当てられた結果、そのうち 6 つのジョブの実行が終了している。これにより、ジョブを実行中の資源ノード 1 0 2 の価値 V_a が $V_a = 4$ から $V_a = 6$ に引き上げられている。

【 0 1 3 7 】

すなわち、投入されたジョブのうち実行が完了したジョブの割合（8分の6）が大きくなったため、ジョブの実行が完了していない資源ノード 1 0 2 の価値 V_a が引き上げられている。

【 0 1 3 8 】

50

この図 2 2 ~ 図 2 5 のブローカリング処理によれば、投入されたジョブの集合のうち実行が完了したジョブの割合に応じて、ジョブを実行中の資源ノード 1 0 2 の優先度を高く設定することができる。これにより、現在実行中のジョブについてより確実に実行を完了させることができ、全体ジョブの実行を早期に完了させることができる。

【 0 1 3 9 】

(優先度設定の具体例(その4))

つぎに、あるサービスについてジョブを実行している際に、他のサービスについてのジョブに割り当てられた回数に応じて資源ノード 1 0 2 の優先度を設定する場合について説明する。

【 0 1 4 0 】

あるサービスについてジョブを実行中の資源ノード 1 0 2 を他のサービスに切り替えた場合、これまでのジョブが無駄となってしまうことがある。このため、他のサービスに切り替えられる回数が多いジョブについて優先度を高く設定することにより、ジョブの実行が終了する前に他のサービスに切り替えられてしまうことを防ぐ。

【 0 1 4 1 】

ここで、他のサービスについてのジョブに割り当てられた回数に応じて資源ノード 1 0 2 の優先度を設定する場合の資源ブローカリング処理の具体例について図 2 6 ~ 図 2 9 を用いて説明する。図 2 6 ~ 図 2 9 は、実施例 2 における資源ブローカリング処理のさらに他の具体例を示す説明図である。

【 0 1 4 2 】

図 2 6 において、サービス A についてジョブを実行している資源ノード 1 0 2 があり、その価値 V_a は $V_a = 1$ である。ここで、現在サービス A についてジョブを実行中のこの資源ノード 1 0 2 を、他のサービスについてのジョブに割り当てる。

【 0 1 4 3 】

図 2 7 において、サービス A についてジョブを実行中の資源ノード 1 0 2 が、他のサービスについてのジョブに割り当てられたため、資源ノード 1 0 2 の価値 V_a が $V_a = 1$ から $V_a = 5$ に引き上げられている。このとき、サービス A についてのジョブの実行が終了していないため、継続して資源ノード 1 0 2 を要求する(図 2 7 中、点線の丸図形)。

【 0 1 4 4 】

図 2 8 において、サービス A についてジョブに資源ノード 1 0 2 を割り当て、中断していたジョブの実行を再試行する。このとき、サービス A についてジョブを実行している資源ノード 1 0 2 の価値 V_a は $V_a = 5$ である。ここで、現在サービス A についてジョブを実行中のこの資源ノード 1 0 2 を、他のサービスについてのジョブに割り当てる。

【 0 1 4 5 】

図 2 9 において、サービス A についてジョブを実行中の資源ノード 1 0 2 が、他のサービスについてのジョブに割り当てられたため、資源ノード 1 0 2 の価値 V_a が $V_a = 5$ から $V_a = 10$ に引き上げられている。このとき、サービス A についてのジョブの実行が終了していないため、継続して資源ノード 1 0 2 を要求する(図 2 9 中、点線の丸図形)。

【 0 1 4 6 】

この図 2 6 ~ 図 2 9 のブローカリング処理によれば、ジョブの実行の再試行がおこなわれるごとに、資源ノード 1 0 2 の優先度を高く設定することができる。このため、再試行の回数が多い資源ノード 1 0 2 を他のサービスへの切替対象外とすることができ、結果的にリソーススタベーションを回避することができる。

【 0 1 4 7 】

以上説明したように、資源ブローカリングプログラム、該プログラムを記録した記録媒体、資源ブローカリング装置、および資源ブローカリング方法によれば、効率的なブローカリングをおこなうことにより、個々のサービスの円滑な提供を実現することができる。

【 0 1 4 8 】

なお、本実施の形態で説明した資源ブローカリング方法は、予め用意されたプログラムをパーソナル・コンピュータやワークステーションなどのコンピュータで実行することに

10

20

30

40

50

より実現することができる。このプログラムは、ハードディスク、フレキシブルディスク、CD-ROM、MO、DVDなどのコンピュータで読み取り可能な記録媒体に記録され、コンピュータによって記録媒体から読み出されることによって実行される。またこのプログラムは、インターネットなどのネットワークを介して配布することが可能な伝送媒体であってもよい。

【0149】

(付記1) サービスごとに当該サービスに利用される資源ノードの価値を所定の価値基準上において相対的に設定させる設定工程と、

前記複数のサービスのうちのサービスの要求を受け付けさせる受付工程と、

前記受付工程によって要求が受け付けられた一のサービスに利用される資源ノードが不足しているか否かを検出させる検出工程と、

前記検出工程によって不足していると検出された場合、前記設定工程によって設定された前記一のサービスに利用される資源ノードの価値が、前記設定工程によって設定された前記一のサービス以外の他のサービスに利用されている資源ノードの価値よりも高いか否かを判断させる判断工程と、

前記判断工程によって判断された判断結果に基づいて、前記他のサービスに利用されている資源ノードを、前記一のサービスに利用される資源ノードに切り替えさせる切替工程と、

をコンピュータに実行させることを特徴とする資源ブローカリングプログラム。

【0150】

(付記2) 前記設定工程は、

前記他のサービスに利用される資源ノードの価値が、当該資源ノードの状態に基づいて変動するように設定させることを特徴とする付記1に記載の資源ブローカリングプログラム。

【0151】

(付記3) 前記設定工程は、

前記他のサービスに利用される資源ノードの価値が、当該資源ノードの実行時間に応じて大きくなるように設定させることを特徴とする付記2に記載の資源ブローカリングプログラム。

【0152】

(付記4) 前記設定工程は、

前記他のサービスに利用される資源ノードの価値が、当該資源ノードの処理性能に応じて変動するように設定させることを特徴とする付記2に記載の資源ブローカリングプログラム。

【0153】

(付記5) 前記設定工程は、

前記他のサービスに利用される資源ノードの価値が、当該資源ノードの使用頻度に応じて大きくなるように設定させることを特徴とする付記2に記載の資源ブローカリングプログラム。

【0154】

(付記6) 前記設定工程は、

前記他のサービスに利用される資源ノードの価値が、当該資源ノードの設置位置に応じて変動するように設定させることを特徴とする付記2に記載の資源ブローカリングプログラム。

【0155】

(付記7) 前記設定工程は、

前記他のサービスに利用される資源ノードの価値が、指定入力された当該資源ノードの実行時間に応じて変動するように設定させることを特徴とする付記2に記載の資源ブローカリングプログラム。

【0156】

(付記 8) 前記設定工程は、

前記他のサービスに利用される資源ノードの価値が、指定入力された実行時間内における当該資源ノードの中断または終了の可否に応じて変動するように設定させることを特徴とする付記 7 に記載の資源ブローカリングプログラム。

【 0 1 5 7 】

(付記 9) 前記設定工程は、

前記他のサービスに関するジョブの集合のうち実行が完了したジョブの割合に応じて、前記他のサービスに利用される資源ノードのうち前記ジョブの実行が完了していない資源ノードの価値が大きくなるように設定させることを特徴とする付記 2 に記載の資源ブローカリングプログラム。

【 0 1 5 8 】

(付記 10) 前記設定工程は、

前記他のサービスに利用される資源ノードの価値が、当該資源ノードの切替回数に応じて大きくなるように設定させることを特徴とする付記 2 に記載の資源ブローカリングプログラム。

【 0 1 5 9 】

(付記 11) 前記切替工程は、

前記判断工程によって判断された判断結果に基づいて、前記他のサービスに利用されている資源ノードのうち最低価値の資源ノードを、前記一のサービスに利用される資源ノードに切り替えさせることを特徴とする付記 1 ~ 10 のいずれか一つに記載の資源ブローカリングプログラム。

【 0 1 6 0 】

(付記 12) 付記 1 ~ 11 のいずれか一つに記載の資源ブローカリングプログラムを記録した前記コンピュータに読み取り可能な記録媒体。

【 0 1 6 1 】

(付記 13) サービスごとに当該サービスに利用される資源ノードの価値を所定の価値基準上において相対的に設定する設定手段と、

前記複数のサービスのうちのサービスの要求を受け付ける受付手段と、

前記受付手段によって要求が受け付けられた一のサービスに利用される資源ノードが不足しているか否かを検出する検出手段と、

前記検出手段によって不足していると検出された場合、前記設定手段によって設定された前記一のサービスに利用される資源ノードの価値が、前記設定手段によって設定された前記一のサービス以外の他のサービスに利用されている資源ノードの価値よりも大きいかなかを判断する判断手段と、

前記判断手段によって判断された判断結果に基づいて、前記他のサービスに利用されている資源ノードを、前記一のサービスに利用される資源ノードに切り替える切替手段と、

を備えることを特徴とする資源ブローカリング装置。

【 0 1 6 2 】

(付記 14) 前記設定手段は、

前記他のサービスに利用される資源ノードの価値が、当該資源ノードの状態に基づいて変動するように設定することを特徴とする付記 13 に記載の資源ブローカリング装置。

【 0 1 6 3 】

(付記 15) 前記切替手段は、

前記判断手段によって判断された判断結果に基づいて、前記他のサービスに利用されている資源ノードのうち最低価値の資源ノードを、前記一のサービスに利用される資源ノードに切り替えることを特徴とする付記 13 または 14 に記載の資源ブローカリング装置。

【 0 1 6 4 】

(付記 16) サービスごとに当該サービスに利用される資源ノードの価値を所定の価値基準上において相対的に設定する設定工程と、

前記複数のサービスのうちのサービスの要求を受け付ける受付工程と、

10

20

30

40

50

前記受付工程によって要求が受け付けられた一のサービスに利用される資源ノードが不足しているか否かを検出する検出工程と、

前記検出工程によって不足していると判断された場合、前記設定工程によって設定された前記一のサービスに利用される資源ノードの価値が、前記設定工程によって設定された前記一のサービス以外の他のサービスに利用されている資源ノードの価値よりも高いか否かを判断する判断工程と、

前記判断工程によって判断された判断結果に基づいて、前記他のサービスに利用されている資源ノードを、前記一のサービスに利用される資源ノードに切り替える切替工程と、
を含んだことを特徴とする資源ブローカリング方法。

【0165】

(付記17)前記設定工程は、

前記他のサービスに利用される資源ノードの価値が、当該資源ノードの状態に基づいて変動するように設定することを特徴とする付記16に記載の資源ブローカリング方法。

【0166】

(付記18)前記切替工程は、

前記判断工程によって判断された判断結果に基づいて、前記他のサービスに利用されている資源ノードのうち最低価値の資源ノードを、前記一のサービスに利用される資源ノードに切り替えることを特徴とする付記16または17に記載の資源ブローカリング方法。

【産業上の利用可能性】

【0167】

以上のように、本発明にかかる資源ブローカリングプログラム、該プログラムを記録した記録媒体、資源ブローカリング装置、および資源ブローカリング方法は、複数のサービス間で利用される資源ノードをブローカリングするシステムに有用である。

【図面の簡単な説明】

【0168】

【図1】この発明の実施の形態にかかる資源ブローカリングシステムのシステム構成図である。

【図2】図1に示したコンピュータ装置のハードウェア構成を示すブロック図である。

【図3】この発明の実施の形態にかかる資源ブローカリング装置の機能的構成を示すブロック図である。

【図4】この発明の実施の形態にかかる資源ブローカリング装置の資源ブローカリング処理手順を示すフローチャートである。

【図5】資源ブローカリング処理の具体例を示す説明図(その1)である。

【図6】資源ブローカリング処理の具体例を示す説明図(その2)である。

【図7】資源ブローカリング処理の具体例を示す説明図(その3)である。

【図8】資源ブローカリング処理の具体例を示す説明図(その4)である。

【図9】資源ブローカリング処理の具体例を示す説明図(その5)である。

【図10】資源ブローカリング処理の具体例を示す説明図(その6)である。

【図11-1】サービスAにおける資源ブローカリング処理の適用前のサービス処理状態を示すグラフである。

【図11-2】サービスAにおける資源ブローカリング処理の適用後のサービス処理状態を示すグラフである。

【図11-3】サービスBにおける資源ブローカリング処理の適用前のサービス処理状態を示すグラフである。

【図11-4】サービスBにおける資源ブローカリング処理の適用後のサービス処理状態を示すグラフである。

【図12-1】資源ブローカリング処理の適用前における資源ノードの割り当て状態を示すグラフである。

【図12-2】資源ブローカリング処理の適用後における資源ノードの割り当て状態を示すグラフである。

10

20

30

40

50

【図 1 3】資源ブローカリングシステムの詳細なシステム構成図である。

【図 1 4】実施例 1 における資源ブローカリング処理を示すシーケンス図である。

【図 1 5】実施例 2 における資源ブローカリング処理の具体例を示す説明図（その 1）である。

【図 1 6】実施例 2 における資源ブローカリング処理の具体例を示す説明図（その 2）である。

【図 1 7】実施例 2 における資源ブローカリング処理の具体例を示す説明図（その 3）である。

【図 1 8】実施例 2 における資源ブローカリング処理の具体例を示す説明図（その 4）である。

10

【図 1 9】テレビ会議システムにおける優先度設定の概要を示す説明図である。

【図 2 0】コンピュータ装置に表示される G U I の一例を示す説明図である。

【図 2 1】バッチ処理システムにおける優先度設定の概要を示す説明図である。

【図 2 2】実施例 2 における資源ブローカリング処理の他の具体例を示す説明図（その 1）である。

【図 2 3】実施例 2 における資源ブローカリング処理の他の具体例を示す説明図（その 2）である。

【図 2 4】実施例 2 における資源ブローカリング処理の他の具体例を示す説明図（その 3）である。

【図 2 5】実施例 2 における資源ブローカリング処理の他の具体例を示す説明図（その 4）である。

20

【図 2 6】実施例 2 における資源ブローカリング処理のさらに他の具体例を示す説明図（その 1）である。

【図 2 7】実施例 2 における資源ブローカリング処理のさらに他の具体例を示す説明図（その 2）である。

【図 2 8】実施例 2 における資源ブローカリング処理のさらに他の具体例を示す説明図（その 3）である。

【図 2 9】実施例 2 における資源ブローカリング処理のさらに他の具体例を示す説明図（その 4）である。

【符号の説明】

30

【 0 1 6 9 】

1 0 0 資源ブローカリングシステム

1 0 1 資源ブローカリング装置

1 0 2 資源ノード

3 0 1 設定部

3 0 2 受付部

3 0 3 検出部

3 0 4 判断部

3 0 5 切替部

1 3 0 1 グリッド・サービスサブシステム

40

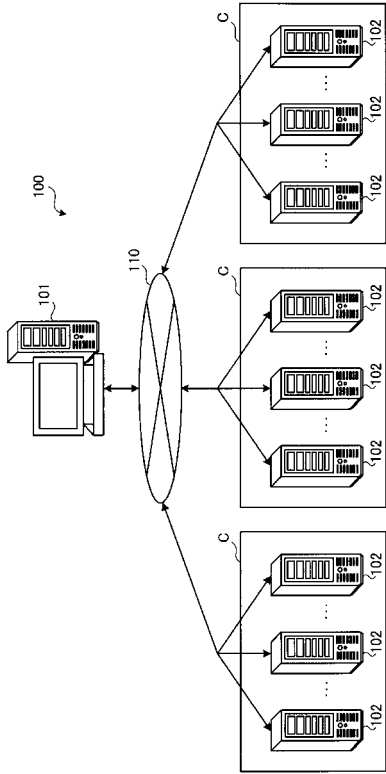
1 3 0 2 資源ブローカリングサブシステム

1 3 0 3 グリッド情報サブシステム

1 3 0 4 運用管理サブシステム

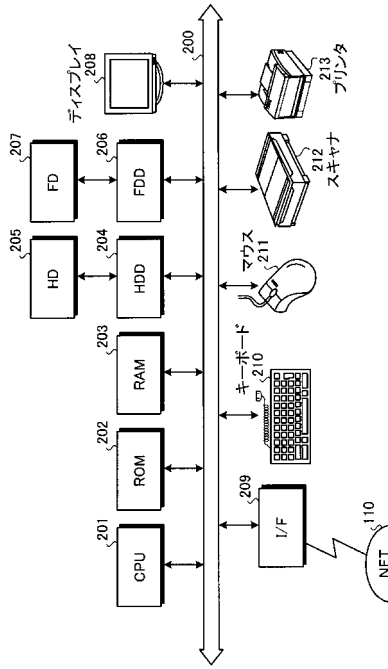
【図1】

この発明の実施の形態にかかる資源ブローカリングシステムのシステム構成図



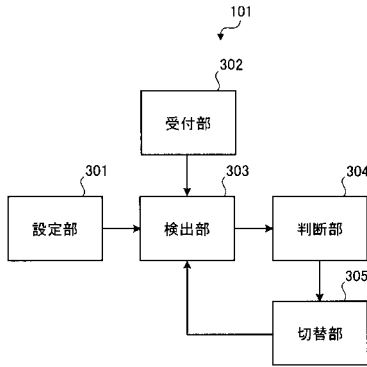
【図2】

図1に示したコンピュータ装置のハードウェア構成を示すブロック図



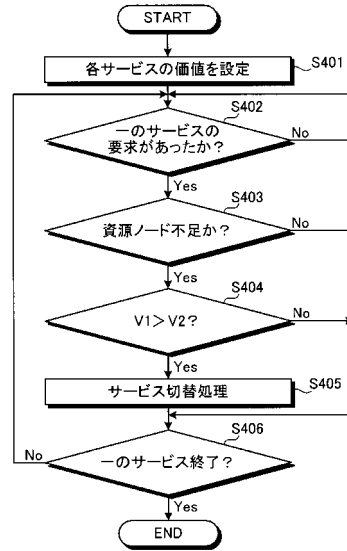
【図3】

この発明の実施の形態にかかる資源ブローカリング装置の機能的構成を示すブロック図



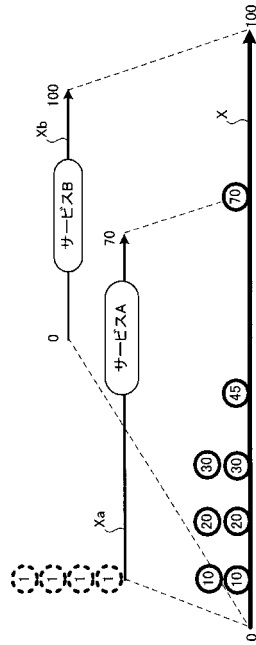
【図4】

この発明の実施の形態にかかる資源ブローカリング装置の資源ブローカリング処理手順を示すフローチャート



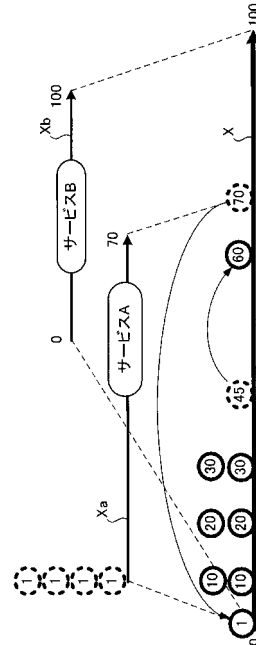
【図5】

資源フローカーリング処理の具体例を示す説明図(その1)



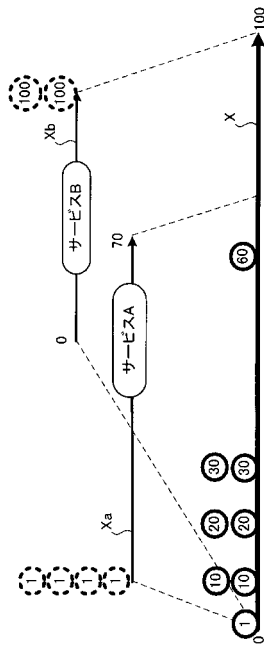
【図6】

資源フローカーリング処理の具体例を示す説明図(その2)



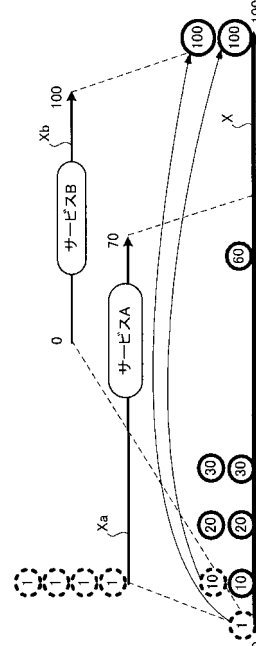
【図7】

資源フローカーリング処理の具体例を示す説明図(その3)

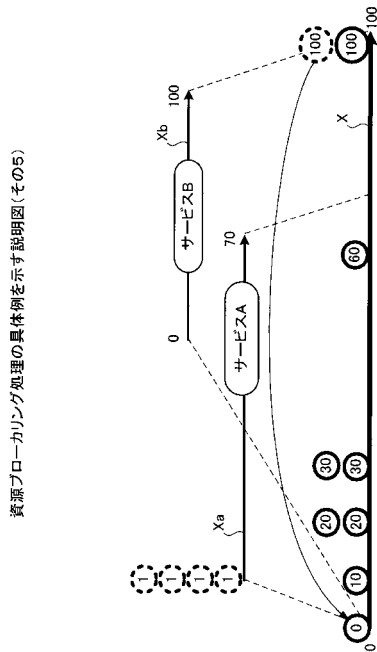


【図8】

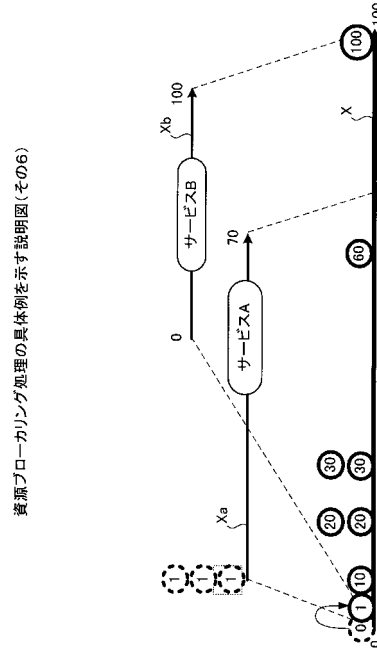
資源フローカーリング処理の具体例を示す説明図(その4)



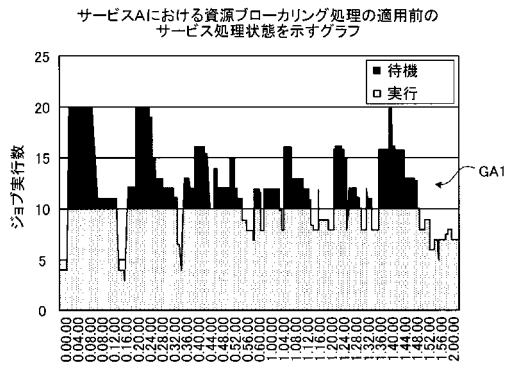
【図9】



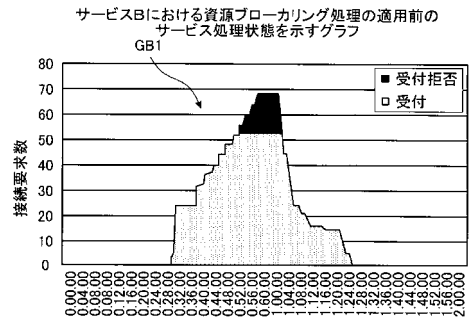
【図10】



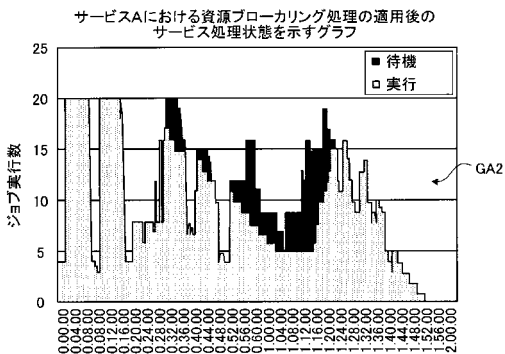
【図11-1】



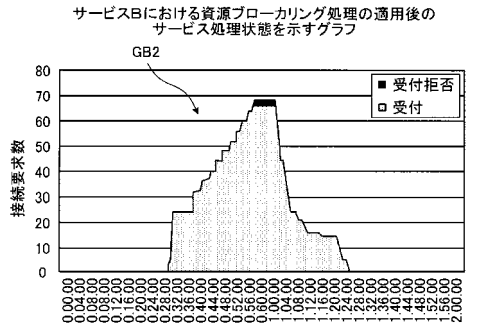
【図11-3】



【図11-2】

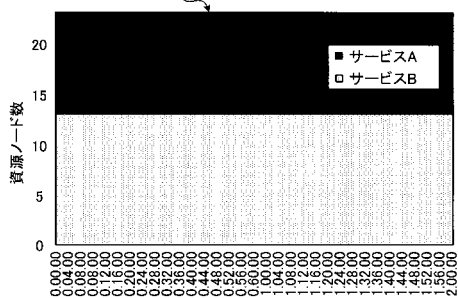


【図11-4】



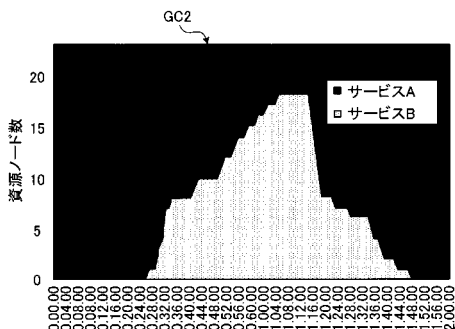
【図12-1】

資源ブローカリング処理の適用前における資源ノードの割り当て状態を示すグラフ



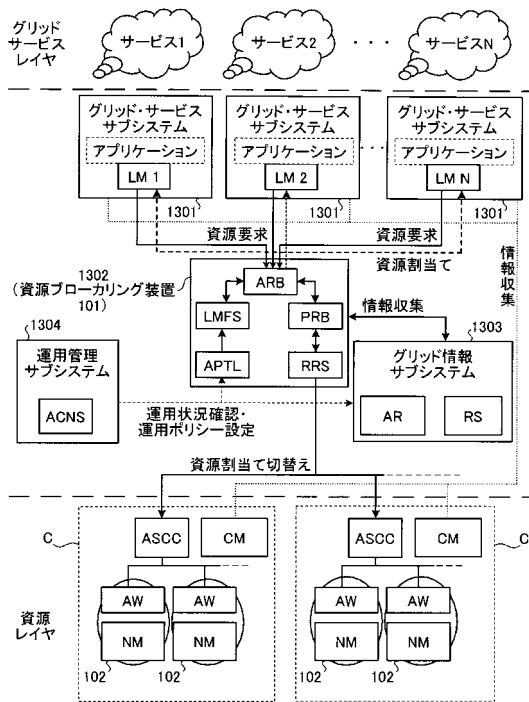
【図12-2】

資源ブローカリング処理の適用後における資源ノードの割り当て状態を示すグラフ



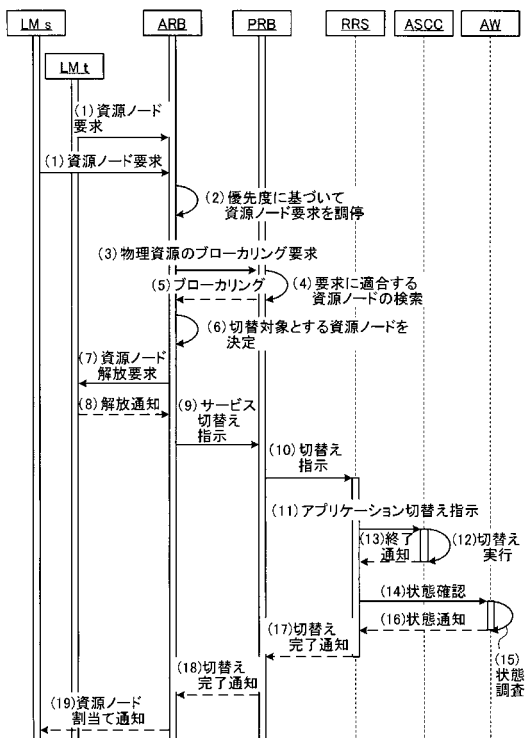
【図13】

資源ブローカリングシステムの詳細なシステム構成図



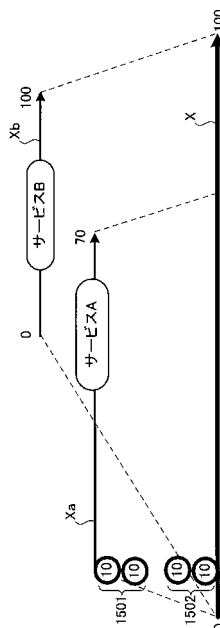
【図14】

実施例1における資源ブローカリング処理を示すシーケンス図

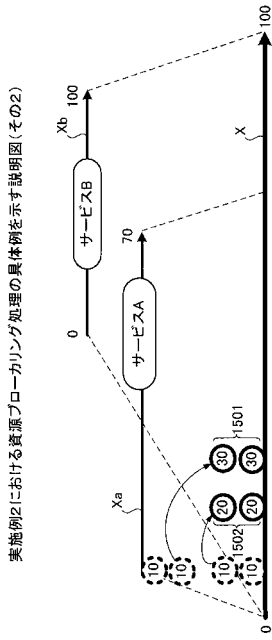


【図15】

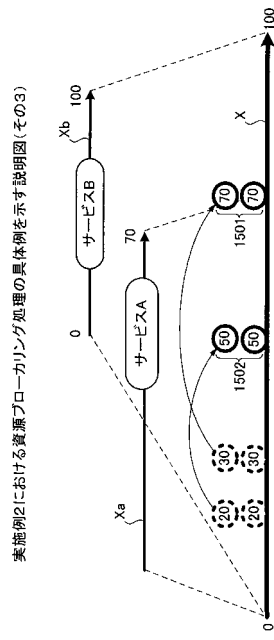
実施例2における資源ブローカリング処理の具体例を示す説明図(その1)



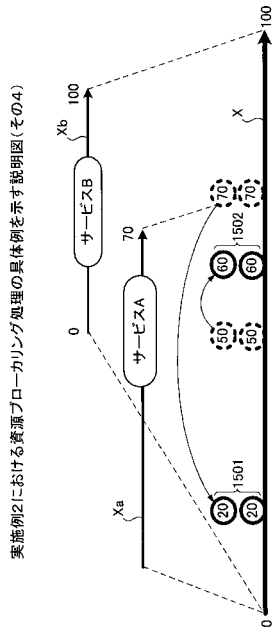
【図16】



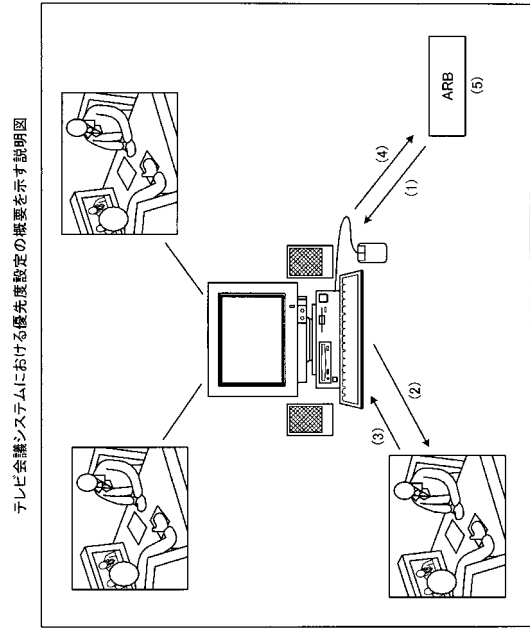
【図17】



【図18】

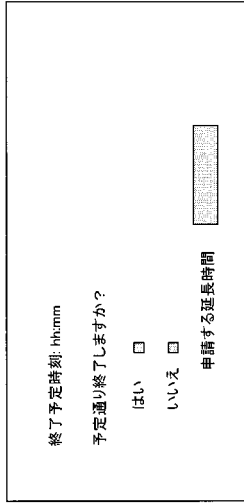


【図19】



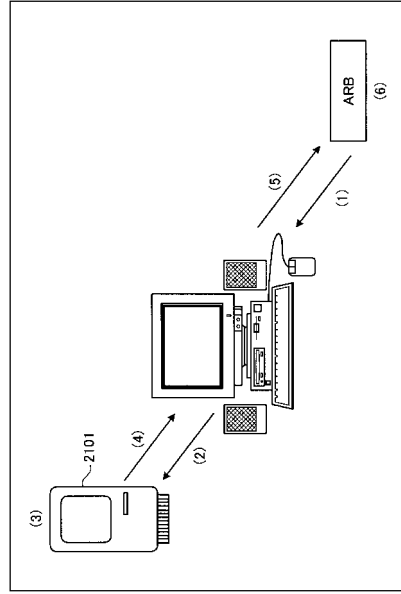
【図20】

コンピュータ装置に表示されるGUIの一例を示す説明図



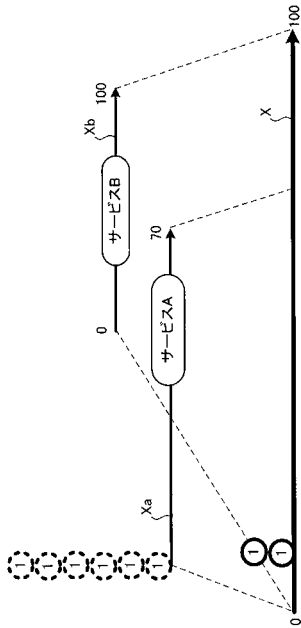
【図21】

バッチ処理システムにおける優先度設定の概要を示す説明図



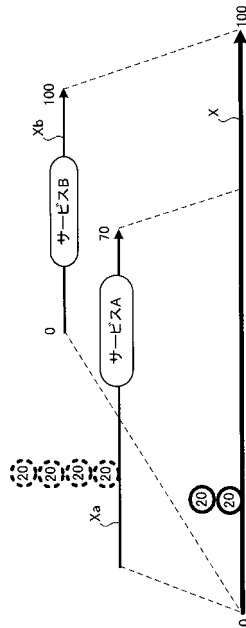
【図22】

実施例2における資源プローカリング処理の他の具体例を示す説明図(その1)

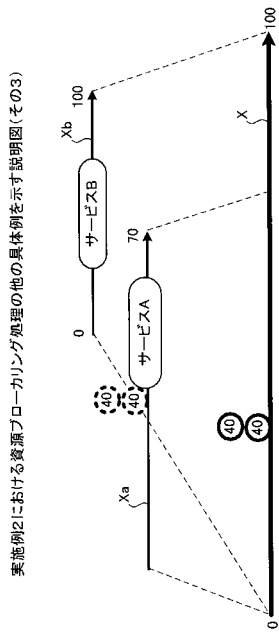


【図23】

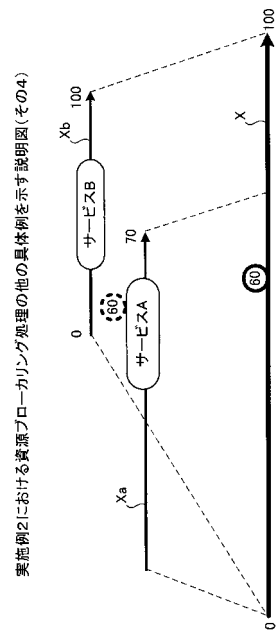
実施例2における資源プローカリング処理の他の具体例を示す説明図(その2)



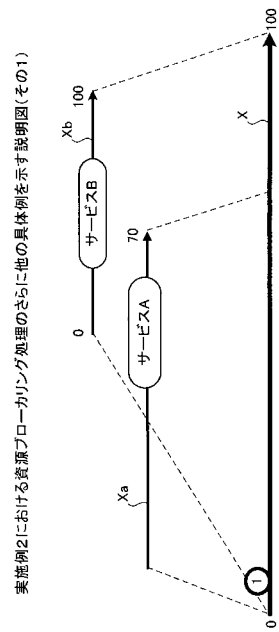
【図 24】



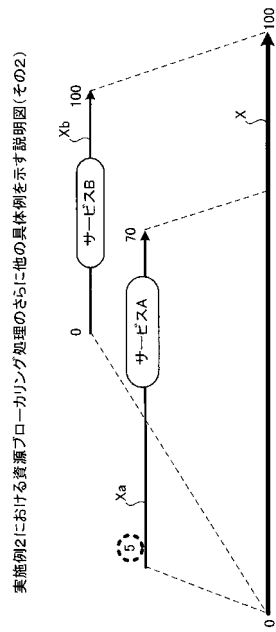
【図 25】



【図 26】

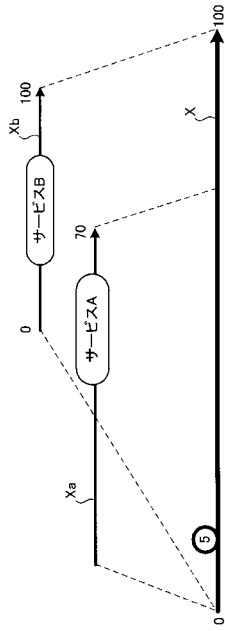


【図 27】



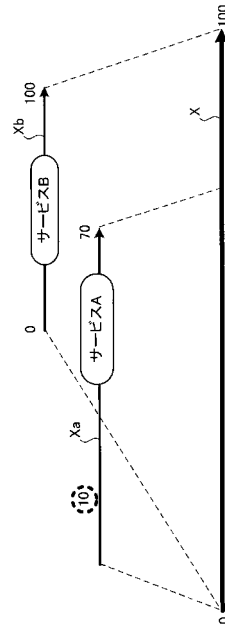
【図 28】

実施例2における資源プロローカリング処理のさらに他の具体例を示す説明図(その3)



【図 29】

実施例2における資源プロローカリング処理のさらに他の具体例を示す説明図(その4)



フロントページの続き

(72)発明者 清水 智弘

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

審査官 鈴木 修治

(56)参考文献 特開平08-022396(JP,A)

特開2003-067351(JP,A)

特表2004-521411(JP,A)

特開平11-085707(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 9/46 - 9/54