(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第5176558号
(P5176558)

(45)発行日　平成25年4月3日(2013.4.3)　(24)登録日　平成25年1月18日(2013.1.18)

(51)Int.Cl.　F I
G06F 9/50 (2006.01)　G06F 9/46 465A

請求項の数 6　(全 25 頁)

(21)出願番号　特願2008-8355(P2008-8355)
(22)出願日　平成20年1月17日(2008.1.17)
(65)公開番号　特開2009-169756(P2009-169756A)
(43)公開日　平成21年7月30日(2009.7.30)
審査請求日　平成22年8月20日(2010.8.20)

(73)特許権者　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号
(74)代理人　100104190
弁理士　酒井　昭徳
(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　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　清木　泰

最終頁に続く

(54)【発明の名称】分散処理プログラム、分散処理装置、および分散処理方法

(57)【特許請求の範囲】
【請求項１】
複数の計算機に複数のジョブを分散処理させる分散処理装置を、
前記複数の計算機から前記ジョブの割当先を決定する決定手段、
前記割当先により計測された、前記割当先に先に送信されたジョブ群の処理要求が受信されてから前記先に送信されたジョブ群の実行が開始されるまでの待ち時間、および前記先に送信されたジョブ群の実行時間を受信する受信手段、
前記割当先に先に送信されたジョブ群の処理要求を送信してから、前記先に送信されたジョブ群の処理結果を前記割当先から受信するまでの経過時間を計測する計測手段、
前記受信手段によって受信された前記待ち時間と前記実行時間と、前記計測手段によって計測された前記経過時間と、前記割当先に先に送信されたジョブ群を形成するジョブの数とに基づいて、前記割当先との通信にかかる通信時間のパラメータを算出し、前記実行時間と前記先に送信されたジョブ群を形成するジョブの数とに基づいて、ジョブ単位の実行時間を算出し、前記割当先に割り当てるジョブ群の実行時間と前記通信時間との対応情報と、算出した前記通信時間のパラメータと前記算出したジョブ単位の実行時間とに基づいて、前記割当先に割り当てる前記ジョブの数を算出する算出手段、
前記算出手段によって算出された前記ジョブの数に基づいて、前記割当先に割り当てるジョブ群を生成する生成手段、
前記生成手段によって生成された前記ジョブ群の処理要求を、前記割当先に送信する送信手段、

として機能させることを特徴とする分散処理プログラム。
【請求項２】
前記決定手段は、
前記各計算機の使用状態に基づいて、前記複数の計算機から前記割当先を決定することを特徴とする請求項１に記載の分散処理プログラム。
【請求項３】
前記決定手段は、
前記各計算機の処理性能に基づいて、前記複数の計算機から前記割当先を決定することを特徴とする請求項１または２に記載の分散処理プログラム。
【請求項４】
前記決定手段は、
前記各計算機が実行可能なジョブのジョブタイプに基づいて、前記複数の計算機から前記割当先を決定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分散処理プログラム。
【請求項５】
複数の計算機に複数のジョブを分散処理させる分散処理装置であって、
前記複数の計算機から前記ジョブの割当先を決定する決定手段と、
前記割当先により計測された、前記割当先に先に送信されたジョブ群の処理要求が受信されてから前記先に送信されたジョブ群の実行が開始されるまでの待ち時間、および前記先に送信されたジョブ群の実行時間を受信する受信手段と、
前記割当先に先に送信されたジョブ群の処理要求を送信してから、前記先に送信されたジョブ群の処理結果を前記割当先から受信するまでの経過時間を計測する計測手段と、
前記待ち時間と前記実行時間と、計測した前記経過時間と、前記割当先に先に送信されたジョブ群を形成するジョブの数とに基づいて、前記割当先との通信にかかる通信時間のパラメータを算出し、前記実行時間と前記先に送信されたジョブ群を形成するジョブの数とに基づいて、ジョブ単位の実行時間を算出し、前記割当先に割り当てるジョブ群の実行時間と前記通信時間との対応情報と算出した前記通信時間のパラメータと前記算出したジョブ単位の実行時間とに基づいて、前記割当先に割り当てる前記ジョブの数を算出する算出手段と、
算出した前記ジョブの数に基づいて、前記割当先に割り当てるジョブ群を生成する生成手段と、
生成した前記ジョブ群の処理要求を前記割当先に送信する送信手段と、
を備えることを特徴とする分散処理装置。
【請求項６】
複数の計算機に複数のジョブを分散処理させる分散処理装置が、
前記複数の計算機から前記ジョブの割当先を決定し、
前記割当先により計測された、前記割当先に先に送信されたジョブ群の処理要求が受信されてから前記先に送信されたジョブ群の実行が開始されるまでの待ち時間、および前記先に送信されたジョブ群の実行時間を受信し、
前記割当先に先に送信されたジョブ群の処理要求を送信してから、前記先に送信されたジョブ群の処理結果を前記割当先から受信するまでの経過時間を計測し、
受信した前記待ち時間と前記実行時間と、計測した前記経過時間と、前記割当先に先に送信されたジョブ群を形成するジョブの数とに基づいて、前記割当先との通信にかかる通信時間のパラメータを算出し、前記実行時間と前記先に送信されたジョブ群を形成するジョブの数とに基づいて、ジョブ単位の実行時間を算出し、前記割当先に割り当てるジョブ群の実行時間と前記通信時間との対応情報と、算出した前記通信時間のパラメータと前記算出したジョブ単位の実行時間とに基づいて、前記割当先に割り当てる前記ジョブの数を算出し、
算出した前記ジョブの数に基づいて、前記割当先に割り当てるジョブ群を生成し、
生成した前記ジョブ群の処理要求を、前記割当先に送信する、

処理を実行することを特徴とする分散処理方法。
【発明の詳細な説明】
【技術分野】
【０００１】
この発明は、マスタ計算機（以下、単に「マスタ」という）が複数のワーカ計算機（以下、単に「ワーカ」という）に処理群を分散処理させるグリッドコンピューティングにおける分散処理プログラム、分散処理装置、および分散処理方法に関する。
【背景技術】
【０００２】
従来、ネットワークを介して通信可能なマスタ／ワーカ間でやり取りされるジョブの流れでは、まず、マスタがジョブおよびその処理に必要なデータをワーカに投入する。つぎに、投入されたワーカは、ジョブの処理を実行する。そして、そのワーカは、ジョブの処理結果をマスタに返す。マスタは、これらを複数のワーカに対しておこなうことにより、ジョブ全体をワーカに分散処理させている。
【０００３】
一般に、マスタに入力されるジョブは、ジョブ単位の実行時間がマスタ／ワーカ間の通信遅延時間に比べて、十分に大きいジョブ（粗粒度なジョブ）であると想定されている。粗粒度なジョブを分散処理する技術は既に実用化されており、例えば、大量のバッチジョブを分散処理するためのグリッド・ミドルウェアであるＳｙｓｔｅｍ　Ｗａｌｋｅｒ　ＣｙｂｅｒＧＲＩＰ（登録商標）などがある。
【０００４】
また、下記特許文献１には、複数のタスクから構成されるタスクグループ単位で処理の割り当てをおこなう技術が記載されている。具体的には、まず、上位管理装置から下位管理装置にタスクグループの処理を割り当てる。つぎに、下位管理装置からタスク実行装置にタスクグループに含まれるタスクを割り当て、タスク実行装置によりタスクを実行し、その実行結果を下位管理装置に送信する。そして、下位管理装置により、タスクの実行結果を収集し、その収集結果を上位管理装置に送信する。最後に、上位管理装置により、収集結果を集約して出力する。
【０００５】
また、下記特許文献２には、マスタからワーカに割り当てるタスク量を、ワーカからマスタへのタスク要求に応じて動的に調整する技術が記載されている。具体的には、まず、マスタが各ワーカにて処理すべきタスクを割り当てる。このとき、最初に同一量のタスクを各ワーカに割り当て、所定時間内の各ワーカからのタスク要求回数に応じて、一度のタスク要求に対して割り当てるタスク量を変化させる。
【０００６】
【特許文献１】特開２００４－１１０３１８号公報
【特許文献２】特開平１１－１９５００７号公報
【発明の開示】
【発明が解決しようとする課題】
【０００７】
上述した従来技術では、マスタに入力されたジョブをジョブ単位でワーカに投入することで、ジョブ全体をワーカに分散処理させている。これは、ジョブ単位の実行時間がマスタ／ワーカ間の通信時間に比べて十分に大きい場合に有効な手法である。ところが、ジョブ単位の実行時間がマスタ／ワーカ間の通信時間よりも短いジョブを大量に処理することが要求されるアプリケーションもある。
【０００８】
しかしながら、実行時間の短いジョブをジョブ単位でワーカに投入すると、ジョブ処理中におけるマスタ／ワーカ間の通信時間やワーカのアイドル時間などのオーバーヘッドが顕在化してしまう。これでは、ネットワーク上のトラフィックが増大化し、ひいてはジョブ処理にかかる所要時間の長期化を招くという問題がある。

【０００９】
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、実行時間の短いジョブを束ねたジョブ群を適切に割り当てることにより、ジョブ処理中におけるマスタ／ワーカ間の通信トラフィックの低減を図り、効率的な分散処理を実現することができる分散処理プログラム、分散処理装置、および分散処理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】
【００１０】
上述した課題を解決し、目的を達成するため、この分散処理プログラム、分散処理装置、および分散処理方法は、通信可能なワーカ計算機群の中からジョブの割当先を決定し、決定された割当先のワーカ計算機の処理性能と、前記割当先との通信にかかる通信時間とに基づいて、前記割当先に割り当てる前記ジョブのジョブ数を算出し、算出されたジョブ数に基づいて、前記割当先に割り当てるジョブ群を生成し、生成されたジョブ群の処理要求を、前記割当先に送信することを要件とする。
【００１１】
この分散処理プログラム、分散処理装置、および分散処理方法によれば、ジョブ単位の実行時間がマスタ／ワーカ間の通信時間よりも短いジョブを束ねて、１つのジョブ群単位でワーカ群に分散処理させることができる。このとき、マスタ／ワーカ間の通信時間およびワーカの処理性能に応じたジョブ数で束ねることで、ワーカ間における終了時刻を平準化し、全体の所要時間の短縮を図ることができる。
【００１２】
また、この分散処理プログラム、分散処理装置、および分散処理方法において、前記割当先により計測された、前記処理要求よりも先に送信された一のジョブ群の処理要求が受信されてから当該一のジョブ群の実行が開始されるまでの待ち時間、および前記一のジョブ群の実行時間に関する情報を受信し、前記割当先に前記一のジョブ群の処理要求を送信してから、前記一のジョブ群の処理結果を前記割当先から受信するまでの経過時間を計測し、受信された待ち時間および実行時間に関する情報と、計測された経過時間とを用いて、前記通信時間を算出することとしてもよい。
【００１３】
この分散処理プログラム、分散処理装置、および分散処理方法によれば、割当先にジョブ群を割り当てることで得られた待ち時間、実行時間および経過時間に関する過去の計測値を用いて、マスタと割当先との間の通信時間を算出することができる。
【００１４】
また、この分散処理プログラム、分散処理装置、および分散処理方法において、さらに、前記一のジョブ群の実行時間に関する情報を用いて、前記処理性能を算出することとしてもよい。
【００１５】
この分散処理プログラム、分散処理装置、および分散処理方法によれば、割当先にジョブ群を割り当てることで得られた実行時間に関する過去の計測値を用いて、割当先の処理性能を算出することができる。
【００１６】
また、この分散処理プログラム、分散処理装置、および分散処理方法において、前記各ワーカ計算機の使用状態に基づいて、前記ワーカ計算機群の中から前記割当先を決定することとしてもよい。
【００１７】
この分散処理プログラム、分散処理装置、および分散処理方法によれば、管理下にあるワーカ群のうち、マスタから割り当てられたジョブ群を処理中、または、機能を停止中のワーカを排除することで、割当先候補を絞り込むことができる。
【００１８】
また、この分散処理プログラム、分散処理装置、および分散処理方法において、前記各ワーカ計算機の処理性能に基づいて、前記ワーカ計算機群の中から前記割当先を決定する

こととしてもよい。
【００１９】
この分散処理プログラム、分散処理装置、および分散処理方法によれば、ジョブ群を高速に処理可能なワーカを割当先に決定することができる。
【００２０】
また、この分散処理プログラム、分散処理装置、および分散処理方法において、前記各ワーカ計算機が実行可能なジョブのジョブタイプに基づいて、前記ワーカ計算機群の中から前記割当先を決定することとしてもよい。
【００２１】
この分散処理プログラム、分散処理装置、および分散処理方法によれば、ワーカＷが受け入れ可能なジョブのジョブタイプを考慮して、ジョブの割当先を決定することで、ジョブ群を適切なワーカＷに割り当てることができる。
【発明の効果】
【００２２】
この分散処理プログラム、分散処理装置、および分散処理方法によれば、実行時間の短いジョブを束ねたジョブ群を適切に割り当てることにより、ジョブ処理中におけるマスタ／ワーカ間の通信トラフィックの低減を図り、効率的な分散処理を実現することができるという効果を奏する。
【発明を実施するための最良の形態】
【００２３】
以下に添付図面を参照して、この分散処理プログラム、分散処理装置、および分散処理方法の好適な実施の形態を詳細に説明する。なお、本明細書において、分散処理装置とは、グリッドコンピューティングシステムを構成する計算機（マスタまたはワーカ）であり、分散処理プログラムとは、分散処理装置にインストールされたプログラムである。
【００２４】
（実施の形態１）
（グリッドコンピューティングシステムのシステム構成）
まず、本実施の形態にかかるグリッドコンピューティングシステム１００のシステム構成について説明する。図１は、グリッドコンピューティングシステムおよび分散処理装置のシステム構成図である。図１において、グリッドコンピューティングシステム１００は、インターネット、ＬＡＮ、ＷＡＮなどのネットワーク１１０を介して通信可能なマスタＭとワーカＷ１～Ｗｍ群とから構成される。
【００２５】
各ワーカＷ１～Ｗｍは、異なる処理能力を持つこととしてもよく、また、ＯＳやハードウェア・アーキテクチャなど異なる構造を持つこととしてもよい。さらに、ネットワーク１１０の通信品質は、一定または画一的である必要もない。
【００２６】
このグリッドコンピューティングシステム１００では、マスタＭが、実行時間の短いジョブ（例えば、解析用プログラム）を束ねたジョブ群を生成し、そのジョブ群を適切なワーカＷ１～Ｗｍに投入する。そして、ワーカＷ１～Ｗｍが、投入されたジョブ群を実行し、その処理結果をマスタＭに返す。このとき、ジョブの処理結果は、ジョブ単位で返すのではなく、ジョブ群単位で返す。
【００２７】
なお、マスタ／ワーカ間でやり取りされる各ジョブの実行時間は一様である。すなわち、各ジョブの粒度（処理量）は揃っている、あるいは、無視できる程度のばらつきである。さらに、各ジョブの実行時間は、ジョブ処理中におけるマスタ／ワーカ間の通信時間（転送時間）よりも短い時間である。
【００２８】
ここで処理対象となるジョブとしては、例えば、金融機関などのオプション・リスク計算において、モンテカルロ法を用いて確率的にシナリオをシミュレーションする際に数万

～数千万個処理することが要求される、実行時間が短い細粒度なジョブが挙げられる。
【００２９】
このような実行時間の短いジョブを大量に処理するケースでは、ジョブ単位でマスタＭからワーカＷ１～Ｗｍにジョブ投入すると、ジョブ処理中におけるマスタ／ワーカ間の通信時間やワーカＷ１～Ｗｍのアイドル時間といったオーバーヘッドが顕在化してしまう。そこで、複数のジョブを束ねたジョブ群を１つの処理単位とする。
【００３０】
図２は、ワーカＷにおけるジョブ処理過程の概要を示す説明図である。図２において、グラフ２１０は、実行時間の短いジョブを１つの処理単位として各ワーカＷ１～Ｗｍに分散処理させる従来の形態のジョブ処理過程を示している。グラフ２２０は、実行時間の短いジョブを束ねて（図２では、３個）１つの処理単位として各ワーカＷ１～Ｗｍに分散処理させる本実施の形態のジョブ処理過程を示している。
【００３１】
グラフ２１０に示すように、実行時間の短いジョブを１つの処理単位として分散処理させた場合には、ジョブ処理中における、マスタ／ワーカ間の通信時間やワーカＷ１～Ｗｍがつぎのジョブを受け取るまでのアイドル時間などのオーバーヘッドが顕在化してしまう。
【００３２】
一方、グラフ２２０に示すように、複数のジョブを束ねて１つの処理単位として分散処理させた場合には、ジョブ処理中における、マスタ／ワーカ間の通信頻度および通信時間を削減し、上記オーバーヘッドを隠蔽する（全所要時間に対する通信時間およびワーカＷ１～Ｗｍのアイドル時間の割合を小さくする）ことができる。
【００３３】
ここで、図１に示したワーカＷ１を例に挙げて、グリッドコンピューティング１００におけるジョブの投入例を説明する。まず、マスタＭにおいて、入力されたジョブＪ１，Ｊ２，Ｊ３の３個を束ねたジョブ群ＪＧ１を生成する。つぎに、その割当先に決定されたワーカＷ１に生成されたジョブ群ＪＧ１を投入（ネットワーク１１０経由で送信）する。
【００３４】
このあと、ワーカＷ１において、マスタＭから投入されたジョブ群ＪＧ１を実行し、ジョブ群ＪＧ１を構成するすべてのジョブＪ１，Ｊ２，Ｊ３の実行が完了したあと、ジョブＪ１，Ｊ２，Ｊ３の処理結果Ｒ１，Ｒ２，Ｒ３を束ねた処理結果ＲＧ１をマスタＭに返す。
【００３５】
このように、複数のジョブＪ１，Ｊ２，Ｊ３を束ねたジョブ群ＪＧ１を１つの処理単位として分散処理させることにより、ジョブ処理中におけるマスタＭとワーカＷ１との間の通信時間を低減し、通信時間やアイドル時間などのオーバーヘッドを隠蔽する。
【００３６】
さらに、割当先のワーカＷ１で処理された処理結果は、すべてのジョブＪ１，Ｊ２，Ｊ３の処理が完了したあとに、割当先のワーカＷ１からマスタＭに返すことで、ジョブＪ１，Ｊ２，Ｊ３の処理中におけるネットワーク１１０上のトラフィックの低減を図る。
【００３７】
（マスタＭおよびワーカＷのハードウェア構成）
つぎに、実施の形態１にかかるマスタＭおよびワーカＷ１～Ｗｍのハードウェア構成について説明する。なお、以降において、特に指定する場合を除いて「ワーカＷ１～Ｗｍ」を「ワーカＷ」と表記する。図３は、マスタＭおよびワーカＷのハードウェア構成を示すブロック図である。
【００３８】
図３において、マスタＭおよびワーカＷは、ＣＰＵ３０１と、ＲＯＭ３０２と、ＲＡＭ３０３と、ＨＤＤ（ハードディスクドライブ）３０４と、ＨＤ（ハードディスク）３０５と、ＦＤＤ（フレキシブルディスクドライブ）３０６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）３０７と、ディスプレイ３０８と、Ｉ／Ｆ（インターフェース）３０９と、キーボード３１０と、マウス３１１と、スキャナ３１２と、プリンタ３１３とを備えている。また、各構成部は、バス３００によってそれぞれ接続されている。

【００３９】

ここで、ＣＰＵ３０１は、マスタＭおよびワーカＷの全体の制御を司る。ＲＯＭ３０２は、ブートプログラムなどのプログラムを記録している。ＲＡＭ３０３は、ＣＰＵ３０１のワークウェアとして使用される。ＨＤＤ３０４は、ＣＰＵ３０１の制御にしたがってＨＤ３０５に対するデータのリード／ライトを制御する。ＨＤ３０５は、ＨＤＤ３０４の制御で書き込まれたデータを記憶する。

【００４０】

ＦＤＤ３０６は、ＣＰＵ３０１の制御にしたがってＦＤ３０７に対するデータのリード／ライトを制御する。ＦＤ３０７は、ＦＤＤ３０６の制御で書き込まれたデータを記憶したり、ＦＤ３０７に記憶されたデータをマスタＭおよびワーカＷに読み取らせたりする。

【００４１】

また、着脱可能な記録媒体として、ＦＤ３０７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３０８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３０８には、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。

【００４２】

Ｉ／Ｆ３０９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／Ｆ３０９は、ネットワーク１１０と内部のインターフェースを司り、ネットワーク１１０からのデータの入出力を制御する。Ｉ／Ｆ３０９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。

【００４３】

キーボード３１０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３１１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の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。

【００４４】

スキャナ３１２は、画像を光学的に読み取り、装置内に画像データを読み込む。なお、スキャナ３１２は、ＯＣＲ機能を持たせてもよい。また、プリンタ３１３は、画像データや文書データを印刷する。プリンタ３１３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などを採用することができる。

【００４５】

（ワーカ管理テーブルの記憶内容）

ここで、ジョブの割当先となるワーカＷ１～ＷｍのＩＰアドレスおよび使用状態を特定する場合に用いられるワーカ管理テーブルについて説明する。図４は、ワーカ管理テーブルの記憶内容を示す説明図である。図４において、ワーカ管理テーブル４００には、ワーカ識別子、ＩＰアドレスおよび状態がワーカＷ１～Ｗｍごとに記憶されている。

【００４６】

状態とは、各ワーカＷ１～Ｗｍの使用状態である。この状態は、マスタＭからのジョブ投入前は「空き」となる。そして、マスタＭからジョブが投入されると「空き」から「使用中」に変更される。また、ワーカＷ１～Ｗｍから処理結果が返ってくると「使用中」から「空き」に変更される。さらに、ワーカＷ１～Ｗｍの機能が停止している場合には「停止中」となる。このワーカ管理テーブル４００は、図３に示したＲＡＭ３０３やＨＤ３０

５などの記憶部によりその機能を実現する。
【００４７】
（スループットテーブルの記憶内容）
ここで、ワーカＷ１～Ｗｍの処理性能を特定する場合に用いられるスループットテーブルについて説明する。図５は、スループットテーブルの記憶内容を示す説明図である。図５において、スループットテーブル５００には、ワーカＷ１～Ｗｍごとのスループット値が記憶されている。
【００４８】
スループット値は、各ワーカＷ１～Ｗｍの処理性能を表わす指標であり、単位時間当たりに処理されたジョブ数を表現する。このスループット値は、各ワーカＷ１～ＷｍのＣＰＵ使用率など（例えば、マスタＭとは異なる他のコンピュータ装置から投入されたジョブを処理中）によって動的に変化する。
【００４９】
また、未だジョブの割当先となっていないワーカＷ１～Ｗｍ（例えば、Ｗ２）のスループット値は記憶されていない（図５中「－」）。このスループットテーブル５００は、図３に示したＲＡＭ３０３やＨＤ３０５などの記憶部によりその機能を実現する。
【００５０】
（分散処理装置の機能的構成）
つぎに、分散処理装置の機能的構成について説明する。まず、マスタＭの機能的構成について説明する。図６は、マスタＭの機能的構成を示すブロック図である。図６において、マスタＭは、決定部６０１と、算出部６０２と、生成部６０３と、送信部６０４と、受信部６０５と、計測部６０６と、から構成されている。
【００５１】
これら各機能６０１～６０６は、マスタＭの記憶部に記憶された当該機能６０１～６０６に関するプログラムをＣＰＵに実行させることにより、または、入出力Ｉ／Ｆにより、当該機能を実現することができる。また、各機能６０１～６０６からの出力データは上記記憶部に保持される。また、図６中矢印で示した接続先の機能は、接続元の機能からの出力データを記憶部から読み込んで、当該機能に関するプログラムをＣＰＵに実行させるものとする。
【００５２】
まず、決定部６０１は、ワーカW群の中からジョブの割当先を決定する機能を有する。具体的には、例えば、各ワーカWの使用状態に基づいて、ワーカW群の中から割当先を決定することとしてもよい。例えば、ワーカ管理テーブル４００からワーカWごとの状態を読み出して、使用状態が「空き」のワーカW群の中から割当先を決定する。これは、管理下にあるワーカW群のうち、マスタMから割り当てられたジョブ群を処理中、または、機能を停止中のワーカWを排除することで、割当先候補を絞り込むことを意味する。
【００５３】
また、各ワーカWの処理性能に基づいて、ワーカW群の中から割当先を決定することとしてもよい。例えば、スループットテーブル５００からワーカWごとのスループット値Ｔを読み出して、スループット値Ｔが最大のワーカWを割当先に決定する。これは、ジョブ群を高速に処理可能なワーカWを割当先に決定することを意味する。さらに、使用状態が「空き」のワーカW群の中から、スループット値Ｔが最大のワーカWを割当先に決定してもよく、また、ランダムに割当先を決定してもよい。
【００５４】
算出部６０２は、決定部６０１によって決定された割当先のワーカWの処理性能と、当該割当先との通信にかかる通信時間とに基づいて、割当先に割り当てるジョブのジョブ数を算出する機能を有する。具体的には、例えば、ジョブ数ｎのジョブを束ねたジョブ群の実行時間が、ジョブ処理中におけるマスタＭ／ワーカW間の通信時間の１～２倍程度になるようにジョブ数ｎを算出する。
【００５５】

すなわち、マスタM／ワーカW間でのジョブ群のジョブ処理中における、通信時間やアイドル時間などのオーバーヘッドが顕在化することを回避する。なお、算出部６０２による算出処理の具体例については後述する。

【００５６】

生成部６０３は、算出部６０２によって算出された算出結果に基づいて、割当先に割り当てるジョブ群を生成する機能を有する。具体的には、例えば、ジョブキューにある一連のジョブを算出部６０２によって算出されたジョブ数ｎで束ねることにより１つのジョブ群を生成する。このとき、ジョブ数ｎ分のジョブがジョブキューにない場合は、ジョブキューにあるすべてのジョブを束ねることとなる。あるいは、待ち時間の上限を設定して、ジョブキューにジョブ数ｎ分のジョブが溜まるのを待つこととしてもよい。

【００５７】

各ジョブには固有のジョブＩＤ（例えば、図１に示したＪ１，Ｊ２，Ｊ３）が割り付けられており、さらに、１つのジョブ群を構成する各ジョブには共通のジョブ群ＩＤ（例えば、ＪＧ１）を割り付ける。このようにジョブＩＤおよびジョブ群ＩＤを割り付けることにより、複数のジョブからなるジョブ群を認識することができる。

【００５８】

なお、ジョブキューとは、マスタM内の記憶部に構築されるデータ構造であり、マスタMに入力されたジョブをキューイングする機能を有する。また、分散処理対象となるジョブは、マスタMに直接入力することとしてもよく、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取得することとしてもよい。

【００５９】

送信部６０４は、生成部６０３によって生成されたジョブ群の処理要求を、割当先に送信する機能を有する。具体的には、例えば、決定部６０１によって決定されたワーカWのＩＰアドレスをワーカ管理テーブル４００から読み出して、そのＩＰアドレスを宛先に設定することでジョブ群の処理要求を割当先に送信することができる。また、送信部６０４により処理要求が送信されると、図４に示したワーカ管理テーブル４００の記憶内容のうち、割当先の状態が「空き」から「使用中」に書き換えられる。

【００６０】

ここで、上記算出部６０２による算出処理の具体例について説明する。受信部６０５は、割当先により計測された、送信部６０４によって上記処理要求よりも先に送信された一のジョブ群の処理要求が受信されてから当該一のジョブ群の実行が開始されるまでの待ち時間、および一のジョブ群の実行時間に関する情報を受信する機能を有する。

【００６１】

待ち時間とは、例えば、割当先において、一のジョブ群の処理要求の受信が検出されてから、一のジョブ群の準備（実行に必要となる関数の呼び出しなど）が整うまでの時間である。実行時間とは、一のジョブ群の実行を開始してから、該一のジョブ群を構成するすべてのジョブの実行が完了するまでに要した時間である。受信部６０５によって受信された情報は、例えば、後述するパラメータテーブル７００に記憶される。

【００６２】

また、計測部６０６は、割当先に一のジョブ群の処理要求を送信してから、一のジョブ群の処理結果を割当先から受信するまでの経過時間を計測する機能を有する。具体的には、例えば、一のジョブ群の処理要求が送信された送信日時と、一のジョブ群の処理結果が受信された受信日時とから経過時間を計測することができる。計測部６０６によって計測された計測結果は、例えば、後述するパラメータテーブル７００に記憶される。

【００６３】

そして、算出部６０２は、受信部６０５によって受信された待ち時間および実行時間に関する情報と、計測部６０６によって計測された経過時間とを用いて、通信時間を算出することとしてもよい。すなわち、割当先にジョブ群を割り当てることで得られた待ち時間、実行時間および経過時間に関する過去の計測値を用いて、マスタと割当先との間の通信時間を算出する。

【００６４】
また、算出部６０２は、一のジョブ群の実行時間に関する情報を用いて、割当先の処理性能を算出することとしてもよい。すなわち、割当先にジョブ群を割り当てることで得られた実行時間に関する過去の計測値を用いて、割当先の処理性能を算出する。
【００６５】
ここで、算出部６０２による算出処理に用いられるパラメータテーブルについて説明する。パラメータテーブルは、例えば、マスタＭの管理下にあるワーカＷごとに、ワーカ識別子と関連付けて保持されている。ここでは、あるワーカＷのパラメータテーブルを例に挙げて説明する。図７は、パラメータテーブルの記憶内容を示す説明図である。図７において、パラメータテーブル７００には、マスタ／ワーカ間の通信時間に関係するパラメータｋ，ｃと、ワーカＷに割り当てられたジョブ群$ＪＧ_1$～$ＪＧ_p$ごとにパラメータ情報７００－１～７００－ｐが記憶されている。
【００６６】
具体的には、パラメータ情報７００－１～７００－ｐは、パラメータｎ，ｔ，ｗ，δ，ｋ，ｃを有している。パラメータｎは、ジョブ群に含まれているジョブの数を表わす値である。パラメータｔは、マスタＭにおいて、ジョブ群の処理要求が送信されてからそのジョブ群の処理結果が受信されるまでの経過時間を表わす値である。パラメータｗは、ワーカＷにおいて、ジョブ群の処理要求が受信されてからそのジョブ群の実行が開始されるまでの待ち時間を表わす値である。
【００６７】
パラメータδは、ワーカＷにおいて一のジョブ群の処理に要した実行時間を表わす値である。パラメータｋ，ｃは、マスタＭ／ワーカＷ間の通信時間に関する値である。ここで、マスタＭ／ワーカＷ間の通信時間とは、例えば、マスタＭからワーカＷへの処理要求の送信、およびワーカＷからマスタＭへの処理結果の送信にかかる時間である。
【００６８】
具体的には、例えば、マスタＭ／ワーカＷ間の通信時間は、下記式（１）を用いて求めることができる。ただし、通信時間をＫ、ジョブ群を構成するジョブ数をｎとする。
【００６９】
$$K = k \times n + c \quad \cdots (1)$$
【００７０】
また、パラメータｋ，ｃは、パラメータｔ，ｗ，δを用いて算出することができる。具体的には、例えば、パラメータｋ，ｃは、下記式（２）を用いて求めることができる。
【００７１】
$$t = w + \delta + (k \times n + c) \quad \cdots (2)$$
【００７２】
より詳細に説明すると、上記式（２）を用いて、$n_x \neq n_y$であるｘ，ｙ（ｘ，ｙは自然数）について、パラメータｋ，ｃの連立方程式（下記式（３）および（４））をたてることで、パラメータｋ，ｃを求めることができる。
【００７３】
$$t_x = w_x + \delta_x + (k \times n_x + c) \quad \cdots (3)$$
【００７４】
$$t_y = w_y + \delta_y + (k \times n_y + c) \quad \cdots (4)$$
【００７５】
このあと、パラメータテーブル７００から、$n_x \neq n_y$である２つのパラメータ情報７００－１～７００－ｐを読み出す。そして、読み出したパラメータ情報７００－１～７００－ｐに含まれるパラメータn_x，n_y，t_x，t_y，w_x，w_y，δ_x，δ_yを上記式（３）および（４）に代入して、パラメータｋ，ｃを求める。
【００７６】
なお、$n_x \neq n_y$である２つのパラメータ情報７００－１～７００－ｐを読み出す場合には、現在時刻から遡って最も直近の時刻に記憶された２つのパラメータ情報７００－１～

７００－ｐを読み出すこととしてもよい。

【００７７】

また、パラメータδ，ｎを用いて、１ジョブ当たりの実行時間を表わすパラメータτを算出することができる。具体的には、例えば、パラメータτは、下記式（５）を用いて求めることができる。ただし、ｊは１からｐの自然数である。

【００７８】

【数１】

$$\tau = \frac{\sum_{j=1}^{p} \delta_j}{\sum_{j=1}^{p} n_j} \quad \cdots(5)$$

【００７９】

そして、求めたパラメータτ，ｋ，ｃを下記式（６）に代入することにより、割当先に割り当てる一のジョブ群として束ねるジョブ数ｎを算出することができる。ただし、ｓは、ジョブ数ｎのジョブ群の実行時間が通信時間Ｋのｓ倍であることを示すパラメータである。パラメータｓの値は、任意に設定可能（例えば、ｓ＝１）である。

【００８０】

$$n \times \tau = s\,(k \times n + c) \quad \cdots(6)$$

【００８１】

なお、上記式（６）を用いて算出されたジョブ数ｎがｎ＞０であった場合には、例えば、ｎに数％のゆらぎを持たせたものをジョブ数とすることとしてもよい。また、上記式（６）を用いて算出されたジョブ数ｎがｎ≦０であった場合には、例えば、予め設定されている数（例えば、ｎ＝１）をジョブ数とすることとしてもよい。

【００８２】

上記パラメータｓの値は、例えば、図３に示したキーボード３１０やマウス３１１などをユーザが操作することで、任意に設定可能である。さらに、マスタＭに入力されたジョブの数が多い場合には、パラメータｓの値を大きくし（例えば、ｓ＝３）、ジョブキューにあるジョブの数が少なくなるにつれて、パラメータｓの値を小さくする（例えば、ｓ＝１）こととしてもよい。

【００８３】

また、図５に示したスループットテーブル５００に記憶されているスループット値Ｔは、上記式（５）を用いて求めたパラメータτの逆数（１／τ）、すなわち、単位時間当たりに処理可能なジョブ数によって表現することができる。

【００８４】

さらに、パラメータテーブル７００の記憶内容は、管理下にあるワーカＷから処理結果を受信する都度、更新される。さらに、上記スループット値Ｔは、例えば、パラメータテーブル７００の記憶内容が更新される都度、合わせて更新されることとなる。パラメータテーブル７００を更新する更新処理については後述する。

【００８５】

なお、パラメータｋ，ｃを求めるために必要となるパラメータ情報７００－１～７００－ｐがパラメータテーブル７００に記憶されていない場合には（例えば、初回の分散処理時）、予め設定されている初期値を用いることとしてもよい。つまり、上記パラメータn_x，n_y，t_x，t_y，w_x，w_y，δ_x，δ_yに相当する初期値を上記式（３）および（４）に代入して、パラメータｋ，ｃを求める。

【００８６】

つぎに、ワーカＷの機能的構成について説明する。図８は、ワーカＷの機能的構成を示すブロック図である。図８において、ワーカＷは、受信部８０１と、実行部８０２と、送信部８０３と、計測部８０４と、から構成されている。

【００８７】
これら各機能８０１～８０４は、ワーカＷの記憶部に記憶された当該機能８０１～８０４に関するプログラムをＣＰＵ３０１に実行させることにより、または、入出力Ｉ／Ｆにより、当該機能を実現することができる。また、各機能８０１～８０４からの出力データは上記記憶部に保持される。また、図８中矢印で示した接続先の機能は、接続元の機能からの出力データを記憶部から読み込んで、当該機能に関するプログラムをＣＰＵに実行させるものとする。
【００８８】
まず、受信部８０１は、ジョブ群の処理要求をマスタＭから受信する機能を有する。また、実行部８０２は、マスタＭから割り当てられたジョブ群の処理を実行する機能を有する。ジョブ群の処理は、ワーカＷのＣＰＵによって実行され、その処理結果はワーカＷ内の記憶部に保持される。
【００８９】
送信部８０３は、実行部８０２によってジョブ群を構成するすべてのジョブの処理の実行が完了した結果、ジョブ群の処理結果をマスタＭに送信する機能を有する。なお、ジョブ群の実行完了は、例えば、以下の手順で検出することができる。具体的には、実行部８０２は、ジョブ群の処理開始時に、予めジョブ群に含まれるジョブ数ｎを送信部８０３に通知する。
【００９０】
さらに、実行部８０２は、ジョブ群に含まれる個々のジョブの実行完了ごとに実行結果を送信部８０３へ通知する。送信部８０３は、実行部８０２から通知されるジョブの実行結果の個数を数えることによって該ジョブ群に含まれるすべてのジョブの実行が完了したことを検出することができる。すなわち、実行部８０２から予め通知されたジョブ数ｎに等しい個数の実行結果を受け取ったときに、ジョブ群の実行が完了したことを検出する。
【００９１】
そして、送信部８０３は、すべてのジョブの実行が完了したことが検出された場合、ジョブ群の処理結果をマスタＭに送信する。このとき、ジョブ群の処理要求から特定されるＩＰアドレスを宛先に設定することで処理結果をマスタＭに送信することができる。
【００９２】
計測部８０４は、受信部８０１によってジョブ群の処理要求が受信されてから、実行部８０２によってジョブ群の実行が開始されるまでの待ち時間を計測する機能を有する。具体的には、例えば、処理要求が受信された受信日時と、ジョブ群の実行が開始された開始日時とから待ち時間を計測することができる。
【００９３】
また、計測部８０４は、実行部８０２によるジョブ群の実行時間を計測する機能を有する。具体的には、例えば、ジョブ群の実行が開始された開始日時と、ジョブ群の実行が完了した完了日時とから実行時間を計測することができる。
【００９４】
また、送信部８０３は、計測部８０４によって計測された上記待ち時間または／および実行時間に関する情報をマスタＭに送信する機能を有する。この情報は、マスタＭの受信部６０５で受信され、算出部６０２による算出処理に用いられる。
【００９５】
（分散処理装置の分散処理手順）
つぎに、分散処理装置の分散処理手順について説明する。まず、マスタＭにおける分散処理手順について説明する。図９は、マスタＭにおける分散処理手順の一例を示すフローチャートである。図９のフローチャートにおいて、まず、ジョブキューにジョブがあるか否かを判断する（ステップＳ９０１）。
【００９６】
ここで、ジョブがマスタＭに入力されるのを待って（ステップＳ９０１：Ｎｏ）、ジョブキューにジョブがある場合（ステップＳ９０１：Ｙｅｓ）、決定部６０１により、管理

下にあるワーカW群の中からジョブの割当先を決定する割当先決定処理を実行する（ステップＳ９０２）。

【００９７】

このあと、算出部６０２により、ステップＳ９０２において決定された割当先のワーカWの処理性能と、割当先との通信にかかる通信時間とに基づいて、割当先に割り当てる一のジョブ群として束ねるジョブ数ｎを算出する（ステップＳ９０３）。

【００９８】

つぎに、ジョブキューへのジョブ入力を待つ待ち時間の上限値を設定する（ステップＳ９０４）。このあと、ステップＳ９０３において算出されたジョブ数ｎがジョブキューにある全ジョブ数Ｎ以下であるか否かを判断する（ステップＳ９０５）。

【００９９】

ここで、ジョブ数ｎ≦全ジョブ数Ｎの場合（ステップＳ９０５：Ｙｅｓ）、生成部６０３により、割当先に割り当てるジョブ数ｎのジョブ群を生成する（ステップＳ９０６）。そして、送信部６０４により、ステップＳ９０６において生成されたジョブ群の処理要求を割当先に送信して（ステップＳ９０７）、本フローチャートによる一連の処理を終了する。

【０１００】

また、ステップＳ９０５において、ジョブ数ｎ＞全ジョブ数Ｎの場合（ステップＳ９０５：Ｎｏ）、ステップＳ９０４において設定された待ち時間の上限値に達したか否かを判断する（ステップＳ９０８）。ここで、上限値に達していない場合（ステップＳ９０８：Ｎｏ）は、ステップＳ９０５に戻る。

【０１０１】

一方、上限値に達した場合（ステップＳ９０８：Ｙｅｓ）、生成部６０３により、ジョブキューにある全ジョブ数Ｎのジョブを用いて、割当先に割り当てるジョブ群を生成する（ステップＳ９０６）。

【０１０２】

なお、ステップＳ９０４において設定される待ち時間の上限値は、例えば、予め規定された値（例えば、１分）を設定することとしてもよく、また、複数の上限値候補の中から、ジョブ数ｎに応じた上限値を自動選択することとしてもよい。

【０１０３】

具体的には、例えば、ジョブ数ｎと上限値とを関連付けて表わす上限値テーブルを参照することで、ステップＳ９０３において算出されたジョブ数に応じた上限値を選択し、待ち時間の上限値に設定することとしてもよい。この上限値テーブルは、例えば、マスタＭ内の記憶部に予め保持されている。

【０１０４】

また、ステップＳ９０４において待ち時間の上限値が設定されると、待ち時間の計測が開始され、このあと、ステップＳ９０８において上限値が設定されてからの経過時間が待ち時間の上限値に達したか否かを判断することとしてもよい。

【０１０５】

つぎに、図９に示したステップＳ９０２における割当先決定処理の詳細な処理手順について説明する。図１０は、割当先決定処理手順の一例を示すフローチャートである。図１０において、まず、ワーカ管理テーブル４００に基づいて、未使用（空き）のワーカWがあるか否かを判断する（ステップＳ１００１）。ここで、未使用のワーカWがなかった場合には（ステップＳ１００１：Ｎｏ）、いずれかのワーカWが使用可能となるのを待つ。

【０１０６】

一方、未使用のワーカWがあった場合（ステップＳ１００１：Ｙｅｓ）、スループットテーブル５００に基づいて、未使用のワーカWの中から、スループット値Ｔが最大のワーカWを選択する（ステップＳ１００２）。そして、ステップＳ１００２において選択されたワーカWを割当先に決定し（ステップＳ１００３）、図９に示したステップＳ９０３に移行する。

【０１０７】
つぎに、ワーカＷにおける分散処理手順について説明する。図１１は、ワーカＷにおける分散処理手順の一例を示すフローチャートである。図１１のフローチャートにおいて、まず、受信部８０１により、ジョブ群の処理要求をマスタＭから受信したか否かを判断する（ステップＳ１１０１）。
【０１０８】
ここで、処理要求を受信するのを待って（ステップＳ１１０１：Ｎｏ）、受信した場合（ステップＳ１１０１：Ｙｅｓ）、計測部８０４により、ジョブ群の処理要求が受信されてから、ジョブ群の実行が開始されるまでの待ち時間の計測を開始する（ステップＳ１１０２）。
【０１０９】
このあと、実行部８０２により、マスタＭから割り当てられたジョブ群の処理を実行する（ステップＳ１１０３）。このとき、計測部８０４により、待ち時間の計測を終了するとともに、ジョブ群の実行時間の計測を開始する（ステップＳ１１０４）。
【０１１０】
このあと、ジョブ群を構成するすべてのジョブの実行が完了したことが検出されると（ステップＳ１１０５）、計測部８０４により、ジョブ群の実行時間の計測を終了する（ステップＳ１１０６）。そして、送信部８０３により、計測部８０４による計測結果を含むジョブ群の処理結果をマスタＭに送信して（ステップＳ１１０７）、本フローチャートによる一連の処理を終了する。
【０１１１】
つぎに、マスタＭにおける、図７に示したパラメータテーブル７００の記憶内容を更新する更新処理手順について説明する。図１２は、パラメータテーブルの更新処理手順の一例を示すフローチャートである。図１２のフローチャートにおいて、まず、受信部６０５により、ワーカＷからジョブ群の処理結果を受信したか否かを判断する（ステップＳ１２０１）。
【０１１２】
ここで、ジョブ群の処理結果を受信するのを待って（ステップＳ１２０１：Ｎｏ）、受信された場合（ステップＳ１２０１：Ｙｅｓ）、ジョブ群の処理結果からパラメータｎ，δ，ｗを取得するとともに、計測部６０６による計測結果からパラメータｔを取得する（ステップＳ１２０２）。このとき、上記ジョブ群のジョブ群ＩＤから特定される計測結果からパラメータｔを取得する。
【０１１３】
このあと、ワーカ管理テーブル４００に基づいて、更新対象のエントリがあるか否かを判断する（ステップＳ１２０３）。これは、ステップＳ１２０１において受信された処理結果が、マスタＭの管理下にあるワーカＷからのものか否かを判断するステップである。
【０１１４】
ここで、更新対象のエントリがある場合（ステップＳ１２０３：Ｙｅｓ）、そのエントリのパラメータテーブル７００に記憶済みのパラメータがあるか否かを判断する（ステップＳ１２０４）。これは、ステップＳ１２０１における処理結果の受信が、初回の受信であるか否かを判断するステップである。
【０１１５】
ここで、記憶済みのパラメータがある場合（ステップＳ１２０４：Ｙｅｓ）、算出部６０２により、パラメータテーブル７００の記憶内容、およびステップＳ１２０２において取得されたパラメータｎ，δ，ｗ，ｔを用いて、パラメータｋ，ｃを算出する（ステップＳ１２０５）。
【０１１６】
このあと、ステップＳ１２０２において取得されたパラメータｎ，δ，ｗ，ｔ、およびステップＳ１２０５において算出されたパラメータｋ，ｃを用いてパラメータテーブル７００の記憶内容を更新して（ステップＳ１２０６）、本フローチャートによる一連の処理

を終了する。

【０１１７】

一方、ステップＳ１２０４において、記憶済みのパラメータがない場合（ステップＳ１２０４：Ｎｏ）、ステップＳ１２０２において取得されたパラメータｎ，δ，ｗ，ｔを用いてパラメータテーブル７００の記憶内容を更新して（ステップＳ１２０６）、本フローチャートによる一連の処理を終了する。

【０１１８】

また、ステップＳ１２０３において、更新対象のエントリがない場合（ステップＳ１２０３：Ｎｏ）、更新処理を強制終了するエラー処理を実行して（ステップＳ１２０８）、本フローチャートによる一連の処理を終了する。さらに、ステップＳ１２０６において、パラメータテーブル７００の記憶内容の更新に合わせて、ワーカ管理テーブル４００の記憶内容（スループット値Ｔ）を更新することとしてもよい。

【０１１９】

以上説明したように、実施の形態１によれば、ジョブ単位の実行時間がマスタＭ／ワーカＷ間の通信時間よりも短いジョブを束ねて、ジョブ群単位でワーカ群に分散処理させることができる。このとき、割当先のワーカＷの処理性能に応じたジョブ数で束ねることで、ワーカＷ間における終了時刻を平準化することができる。

【０１２０】

このように、実行時間の短いジョブを束ねたジョブ群を適切にワーカＷに割り当てることで、ジョブ処理中におけるマスタＭ／ワーカＷ間の通信トラフィックの低減を図り、効率的な分散処理を実現することができる。

【０１２１】

ここで、複数のジョブを、処理性能が異なるワーカＷａ，Ｗｂ，Ｗｃに分散処理させた場合の効果を具体的に説明する。図１３は、ワーカＷごとの所要時間を示すガントチャートである。図１３において、（１）は、ワーカＷａ，Ｗｂ，Ｗｃの処理性能（スループット値Ｔ）を考慮することなく、一定の数（ここでは、３個）のジョブを束ねた場合の所要時間を表わしている。

【０１２２】

（２）は、本実施の形態で説明したように、ワーカＷａ，Ｗｂ，Ｗｃの処理性能（スループット値Ｔ）に応じて、束ねるジョブ数を変化させた場合の所要時間を表わしている。（１）では、処理性能が低いワーカＷｂに律速され、全ジョブの終了時刻が遅くなっている。

【０１２３】

一方、（２）では、ワーカＷａ，Ｗｂ，Ｗｃの処理性能に応じて求めたジョブ数からなるジョブ群を割り当てているため、各ワーカＷａ，Ｗｂ，Ｗｃのジョブ処理にかかる所要時間が平準化され、全ジョブの終了時刻が（１）の場合に比べて短縮されている。

【０１２４】

（実施の形態２）

つぎに、実施の形態２にかかる分散処理装置について説明する。実施の形態２では、図１に示したグリッドコンピューティングシステム１００を構成するワーカＷ１～Ｗｍごとに受け入れ可能（実行可能）なジョブタイプが異なる場合（同一の場合を含む）の分散処理について説明する。なお、実施の形態１において説明した箇所と同一箇所については、同一符号を付して図示および説明を省略する。

【０１２５】

まず、実施の形態２にかかるマスタＭの機能的構成について説明する。決定部６０１は、各ワーカが実行可能なジョブのジョブタイプに基づいて、ワーカ群の中から割当先を決定する機能を有する。ジョブタイプとは、例えば、ジョブの実行時に呼び出される関数名で分類されるジョブの種別である。

【０１２６】

より具体的には、例えば、ジョブの実行時に起動されるアプリケーションによってジョ

ブタイプを分類することができる。すなわち、各ワーカWが持つ機能によっては、マスタMから投入されるジョブを実行できる場合と、実行できない場合とがある。そこで、マスタMが、各ワーカWが実行可能なジョブのジョブタイプを判断し、ジョブの適切な割当先を決定する。

【０１２７】

ここで、実施の形態２にかかるマスタMの分散処理手順の概要について説明する。図１４は、分散処理手順の概要を示す説明図である。図１４において、ジョブタイプの異なる３つのジョブＪａ，Ｊｂ，ＪｃがマスタMに入力されたとする。ここでは、ジョブＪａのジョブタイプをＡ、ジョブＪｂのジョブタイプをＢ、ジョブＪｃのジョブタイプをＣと表記する。

【０１２８】

この場合、各ジョブＪａ，Ｊｂ，Ｊｃのジョブタイプを判断して、ジョブタイプごとに分類されたジョブキューＡ，Ｂ，Ｃに対応するジョブＪａ，Ｊｂ，Ｊｃを配置する。ここでは、ジョブＪａがジョブキューＡに配置され、ジョブＪｂがジョブキューＢに配置され、ジョブＪｃがジョブキューＣに配置される。

【０１２９】

このあと、決定部６０１は、各ワーカWが実行可能なジョブのジョブタイプに基づいて、ワーカW群の中から各ジョブキューＡ，Ｂ，Ｃに配置されたジョブＪａ，Ｊｂ，Ｊｃの割当先を決定することとなる。

【０１３０】

さらに、決定部６０１は、あるジョブタイプのジョブを実行可能なワーカWが複数存在する場合には、複数のワーカWの中からスループット値が最大のワーカWを割当先に決定することとしてもよい。このとき、例えば、ジョブタイプＡのジョブの割当先を決定する場合には、スループット値が最大のワーカWを決定し、ジョブタイプＢのジョブの割当先を決定する場合には、スループット値が２以上のワーカWをランダムに決定するなどのポリシーを設定することとしてもよい。

【０１３１】

ここで、各ワーカWが実行可能なジョブのジョブタイプを特定する場合に用いられるワーカ管理テーブルについて説明する。図１５は、ワーカ管理テーブルの記憶内容を示す説明図である。図１５において、ワーカ管理テーブル１５００には、ワーカＷ１～Ｗｍごとに、ＩＰアドレス、状態および実行可能タイプが記憶されている。実行可能タイプとは、各ワーカWが実行可能なジョブのジョブタイプである。

【０１３２】

つぎに、ワーカＷ１～Ｗｍのジョブタイプごとの処理性能を特定する場合に用いられるスループットテーブルについて説明する。図１６は、スループットテーブルの記憶内容を示す説明図である。図１６において、スループットテーブル１６００には、ワーカＷ１～Ｗｍごとに、実行可能なジョブのジョブタイプごとのスループット値が記憶されている。

【０１３３】

ワーカＷｉを例に挙げると、ジョブタイプＡのジョブに関するスループット値Ｔｉａ、ジョブタイプＢのジョブに関するスループット値Ｔｉｂ、およびジョブタイプＣのジョブに関するスループット値Ｔｉｃを有している。

【０１３４】

ここで、決定部６０１による割当先決定処理の具体例について説明する。まず、決定部６０１は、ワーカ管理テーブル１５００からワーカWごとの状態および実行可能タイプを読み出して、ジョブキューにあるジョブを実行可能でかつ使用状態が「空き」であるワーカWをワーカW群の中から特定する。そして、スループットテーブル１６００から特定されたワーカWのスループット値を読み出し、スループット値が最大のワーカWを割当先に決定する。

【０１３５】

（分散処理装置の分散処理手順）

つぎに、実施の形態２にかかる分散処理装置の分散処理手順について説明する。なお、分散処理装置の分散処理手順のうち、決定部６０１による割当先決定処理（図９で示したステップＳ９０２に相当）以外は、実施の形態１と同様のため説明を省略する。図１７は、割当先決定処理手順の他の一例を示すフローチャートである。

【０１３６】

図１７において、まず、ジョブキューにあるジョブのジョブタイプを取得する（ステップＳ１７０１）。このあと、ワーカ管理テーブル１５００に基づいて、ステップＳ１７０１において取得されたジョブタイプのジョブを実行可能なワーカＷがあるか否かを判断する（ステップＳ１７０２）。

【０１３７】

ここで、実行可能なワーカＷがある場合（ステップＳ１７０２：Ｙｅｓ）、ワーカ管理テーブル１５００に基づいて、上記ジョブタイプのジョブを実行可能なワーカＷのうち、未使用（空き）のワーカＷがあるか否かを判断する（ステップＳ１７０３）。ここで、未使用のワーカＷがなかった場合には（ステップＳ１７０３：Ｎｏ）、いずれかのワーカＷが使用可能となるのを待つ。

【０１３８】

一方、未使用のワーカＷがあった場合（ステップＳ１７０３：Ｙｅｓ）、スループットテーブル１６００に基づいて、上記未使用のワーカＷの中から、スループット値Ｔが最大のワーカＷを選択する（ステップＳ１７０４）。そして、選択されたワーカＷを割当先に決定し（ステップＳ１７０５）、図９に示したステップＳ９０３に移行する。

【０１３９】

また、ステップＳ１７０２において、実行可能なワーカＷがなかった場合には（ステップＳ１７０２：Ｎｏ）、入力されたジョブが実行不可能である旨を示すメッセージを出力するエラー処理を実行して（ステップＳ１７０６）、処理を終了する。

【０１４０】

なお、実施の形態２にかかるマスタＭにおける分散処理は、ワーカＭ内に構築されたジョブキューごとに実行することとしてもよい。また、実施の形態１で説明したパラメータテーブル７００（図７参照）は、各ワーカＷが実行可能なジョブのジョブタイプごとに作成することとしてもよい。これにより、各ワーカＷの処理性能（スループット値）を、ジョブタイプごとに算出することができる。

【０１４１】

以上説明したように、実施の形態２によれば、ワーカＷが受け入れ可能なジョブのジョブタイプを考慮して、ジョブの割当先を決定することで、ジョブ群を適切なワーカＷに割り当てることができる。これにより、割り当てられたジョブ群を実行できない場合のエラー処理や、他のワーカＷへのジョブ群の再割り当てにかかる処理などを削減し、マスタＭ／ワーカＷ間の通信トラフィックの低減を図ることができる。

【０１４２】

また、ワーカＷに入力されたジョブをジョブタイプごとに設けたジョブキューに配置することで、同じ関数を読み出すジョブを束ねることとなり、ワーカＷのキャッシュヒット率を向上させることができる。

【０１４３】

なお、本実施の形態で説明した分散処理方法は、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（伝送媒体）を介してコンピュータに配布することが可能な形態であってもよい。

【０１４４】

上述した実施の形態に関し、さらに以下の付記を開示する。

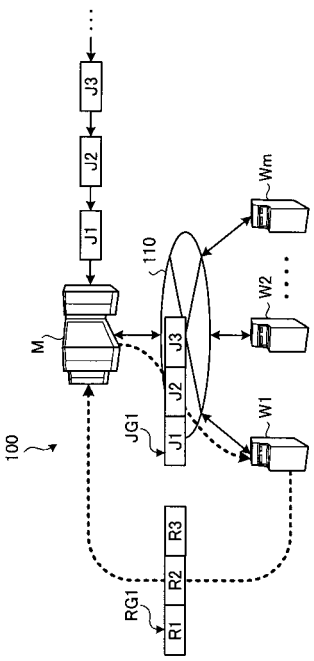
【０１４５】
（付記１）通信可能なワーカ計算機群に複数のジョブを分散処理させるマスタ計算機を、
前記ワーカ計算機群の中から前記ジョブの割当先を決定する決定手段、
前記決定手段によって決定された割当先のワーカ計算機の処理性能と、前記割当先との通信にかかる通信時間とに基づいて、前記割当先に割り当てる前記ジョブのジョブ数を算出する算出手段、
前記算出手段によって算出された算出結果に基づいて、前記割当先に割り当てるジョブ群を生成する生成手段、
前記生成手段によって生成されたジョブ群の処理要求を、前記割当先に送信する送信手段、
として機能させることを特徴とする分散処理プログラム。
【０１４６】
（付記２）前記マスタ計算機を、
前記割当先により計測された、前記送信手段によって前記処理要求よりも先に送信された一のジョブ群の処理要求が受信されてから当該一のジョブ群の実行が開始されるまでの待ち時間、および前記一のジョブ群の実行時間に関する情報を受信する受信手段、
前記割当先に前記一のジョブ群の処理要求を送信してから、前記一のジョブ群の処理結果を前記割当先から受信するまでの経過時間を計測する計測手段として機能させ、
前記算出手段は、
前記受信手段によって受信された待ち時間および実行時間に関する情報と、前記計測手段によって計測された経過時間とを用いて、前記通信時間を算出することを特徴とする付記１に記載の分散処理プログラム。
【０１４７】
（付記３）前記算出手段は、
さらに、前記一のジョブ群の実行時間に関する情報を用いて、前記処理性能を算出することを特徴とする付記２に記載の分散処理プログラム。
【０１４８】
（付記４）前記決定手段は、
前記各ワーカ計算機の使用状態に基づいて、前記ワーカ計算機群の中から前記割当先を決定することを特徴とする付記１～３のいずれか一つに記載の分散処理プログラム。
【０１４９】
（付記５）前記決定手段は、
前記各ワーカ計算機の処理性能に基づいて、前記ワーカ計算機群の中から前記割当先を決定することを特徴とする付記１～４のいずれか一つに記載の分散処理プログラム。
【０１５０】
（付記６）前記決定手段は、
前記各ワーカ計算機が実行可能なジョブのジョブタイプに基づいて、前記ワーカ計算機群の中から前記割当先を決定することを特徴とする付記１～５のいずれか一つに記載の分散処理プログラム。
【０１５１】
（付記７）マスタ計算機と通信可能なワーカ計算機を、
前記マスタ計算機から割り当てられたジョブ群の処理を実行する実行手段、
前記マスタ計算機から前記ジョブ群の処理要求を受信する受信手段、
前記受信手段によって前記処理要求が受信されてから、前記実行手段によって前記ジョブ群の実行が開始されるまでの待ち時間を計測する計測手段、
前記計測手段によって計測された待ち時間に関する情報を前記マスタ計算機に送信する送信手段、
として機能させることを特徴とする分散処理プログラム。
【０１５２】
（付記８）前記計測手段は、

前記実行手段による前記ジョブ群の実行時間を計測し、
前記送信手段は、
前記計測手段によって計測された実行時間に関する情報を前記マスタ計算機に送信することを特徴とする付記７に記載の分散処理プログラム。
【０１５３】
（付記９）通信可能なワーカ計算機群に複数のジョブを分散処理させる分散処理装置であって、
前記ワーカ計算機群の中から前記ジョブの割当先を決定する決定手段と、
前記決定手段によって決定された割当先のワーカ計算機の処理性能と、当該割当先との通信にかかる通信時間とに基づいて、前記割当先に割り当てる前記ジョブのジョブ数を算出する算出手段と、
前記算出手段によって算出された算出結果に基づいて、前記割当先に割り当てるジョブ群を生成する生成手段と、
前記生成手段によって生成されたジョブ群の処理要求を、前記割当先に送信する送信手段と、
を備えることを特徴とする分散処理装置。
【０１５４】
（付記１０）マスタ計算機から割り当てられたジョブ群の処理を実行する実行手段と、
前記マスタ計算機から前記ジョブ群の処理要求を受信する受信手段と、
前記受信手段によって前記処理要求が受信されてから、前記実行手段によって前記ジョブ群の実行が開始されるまでの待ち時間を計測する計測手段と、
前記計測手段によって計測された待ち時間に関する情報を前記マスタ計算機に送信する送信手段と、
を備えることを特徴とする分散処理装置。
【０１５５】
（付記１１）通信可能なワーカ計算機群に複数のジョブを分散処理させる分散処理方法であって、
前記ワーカ計算機群の中から前記ジョブの割当先を決定する決定工程と、
前記決定工程によって決定された割当先のワーカ計算機の処理性能と、当該割当先との通信にかかる通信時間とに基づいて、前記割当先に割り当てる前記ジョブのジョブ数を算出する算出工程と、
前記算出工程によって算出された算出結果に基づいて、前記割当先に割り当てるジョブ群を生成する生成工程と、
前記生成工程によって生成されたジョブ群の処理要求を、前記割当先に送信する送信工程と、
を含んだことを特徴とする分散処理方法。
【０１５６】
（付記１２）マスタ計算機から割り当てられたジョブ群の処理を実行する実行工程と、
前記マスタ計算機から前記ジョブ群の処理要求を受信する受信工程と、
前記受信工程によって前記処理要求が受信されてから、前記実行工程によって前記ジョブ群の実行が開始されるまでの待ち時間を計測する計測工程と、
前記計測工程によって計測された待ち時間に関する情報を前記マスタ計算機に送信する送信工程と、
を含んだことを特徴とする分散処理方法。
【図面の簡単な説明】
【０１５７】
【図１】グリッドコンピューティングシステムおよび分散処理装置のシステム構成図である。
【図２】ワーカＷにおけるジョブ処理過程の概要を示す説明図である。
【図３】マスタＭおよびワーカＷのハードウェア構成を示すブロック図である。

【図４】ワーカ管理テーブルの記憶内容を示す説明図（その１）である。
【図５】スループットテーブルの記憶内容を示す説明図（その１）である。
【図６】マスタＭの機能的構成を示すブロック図である。
【図７】パラメータテーブルの記憶内容を示す説明図である。
【図８】ワーカＷの機能的構成を示すブロック図である。
【図９】マスタＭにおける分散処理手順の一例を示すフローチャートである。
【図１０】割当先決定処理手順の一例を示すフローチャートである。
【図１１】ワーカＷにおける分散処理手順の一例を示すフローチャートである。
【図１２】パラメータテーブルの更新処理手順の一例を示すフローチャートである。
【図１３】ワーカＷごとの所要時間を示すガントチャートである。
【図１４】分散処理手順の概要を示す説明図である。
【図１５】ワーカ管理テーブルの記憶内容を示す説明図（その２）である。
【図１６】スループットテーブルの記憶内容を示す説明図（その２）である。
【図１７】割当先決定処理手順の他の一例を示すフローチャートである。
【符号の説明】
【０１５８】
１００　グリッドコンピューティングシステム
２１０，２２０　グラフ
４００，１５００　ワーカ管理テーブル
５００，１６００　スループットテーブル
６０１　決定部
６０２　算出部
６０３　生成部
６０４　送信部
６０５　受信部
６０６　計測部
７００　パラメータテーブル
７００－１～７００－ｐ　パラメータ情報
８０１　受信部
８０２　実行部
８０３　送信部
８０４　計測部
Ｍ　マスタ
Ｗ，Ｗ１～Ｗｍ　ワーカ

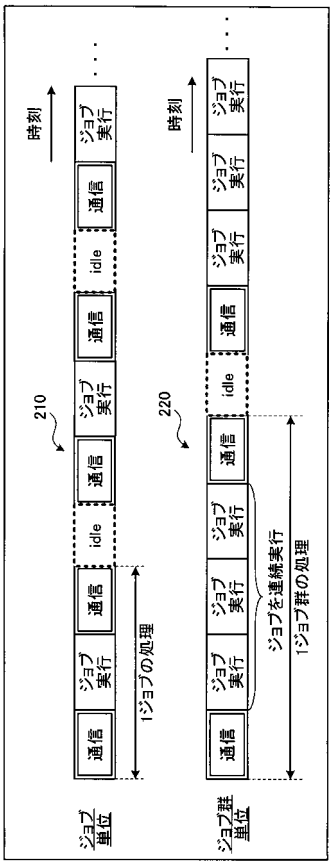
【図１】

グリッドコンピューティングシステムおよび分散処理装置のシステム構成図



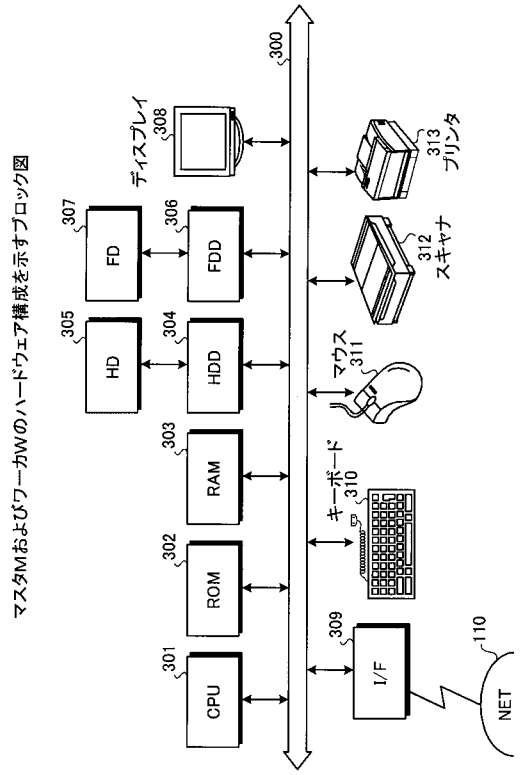
【図２】

ワーカWにおけるジョブ処理過程の概要を示す説明図



【図３】

マスタMおよびワーカWのハードウェア構成を示すブロック図



【図４】

ワーカ管理テーブルの記憶内容を示す説明図（その1）

400

ワーカ識別子	IPアドレス	状態
W1	xxx.yyy.z.t	使用中
W2	xxx.yyy.c.d	停止中
・・・	・・・	・・・
Wi	xxx.yyy.g.j	空き
・・・	・・・	・・・
Wm	xxx.yyy.k.l	使用中

【図５】

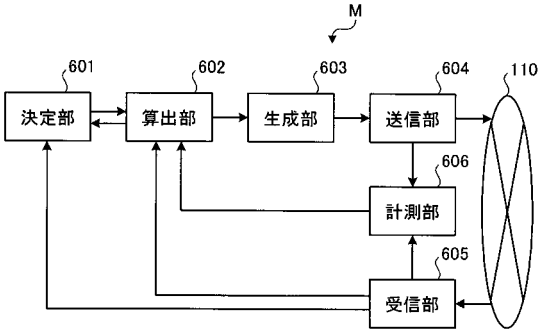
スループットテーブルの記憶内容を示す説明図（その1）

500

ワーカ識別子	スループット値（T）
W1	T1
W2	—
・・・	・・・
Wi	Ti
・・・	・・・
Wm	Tm

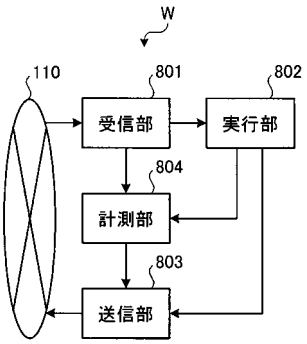
【図６】

マスタMの機能的構成を示すブロック図



【図８】

ワーカWの機能的構成を示すブロック図



【図７】

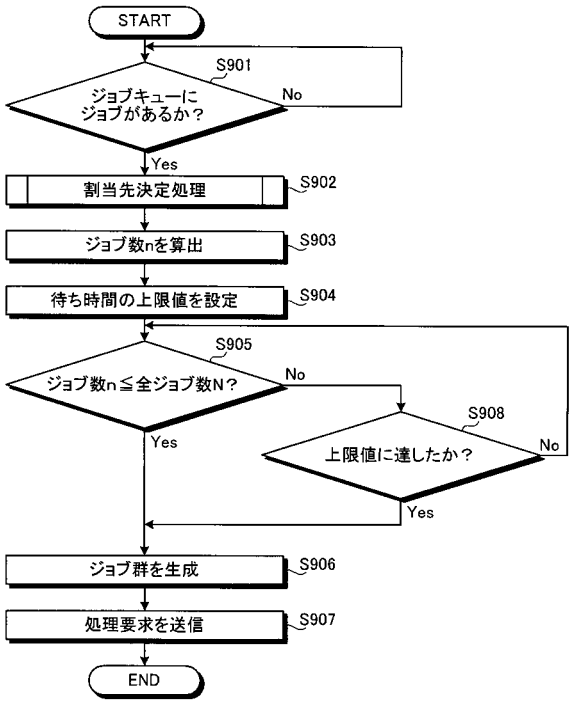
パラメータテーブルの記憶内容を示す説明図

700

	通信時間にかかるパラメータ				
	k	c			
	ジョブ群ID	ジョブ群の実行時間にかかるパラメータ			
		ジョブ数n	経過時間t	待ち時間w	実行時間 δ
700-1	JG_1	n_1	t_1	w_1	δ_1
700-2	JG_2	n_2	t_2	w_2	δ_2
⋮	⋮	⋮	⋮	⋮	⋮
700-j	JG_j	n_j	t_j	w_j	δ_j
⋮	⋮	⋮	⋮	⋮	⋮
700-p	JG_p	n_p	t_p	w_p	δ_p

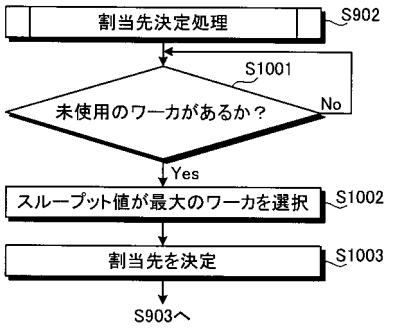
【図９】

マスタMにおける分散処理手順の一例を示すフローチャート



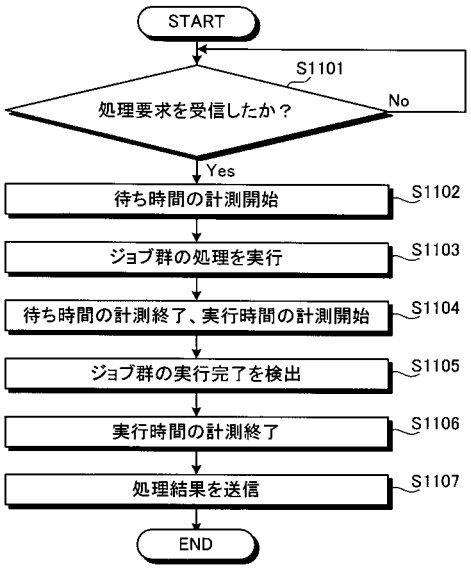
【図１０】

割当先決定処理手順の一例を示すフローチャート



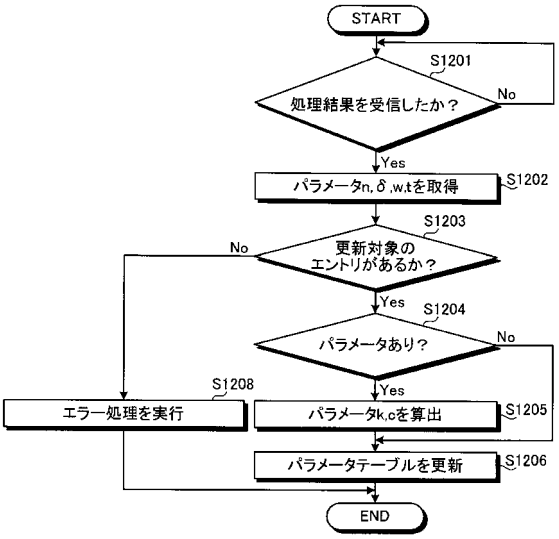
【図１１】

ワーカWにおける分散処理手順の一例を示すフローチャート

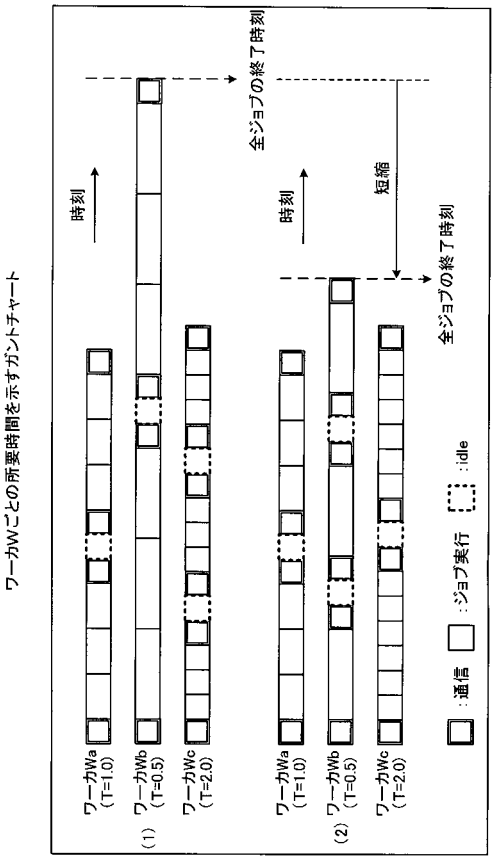


【図１２】

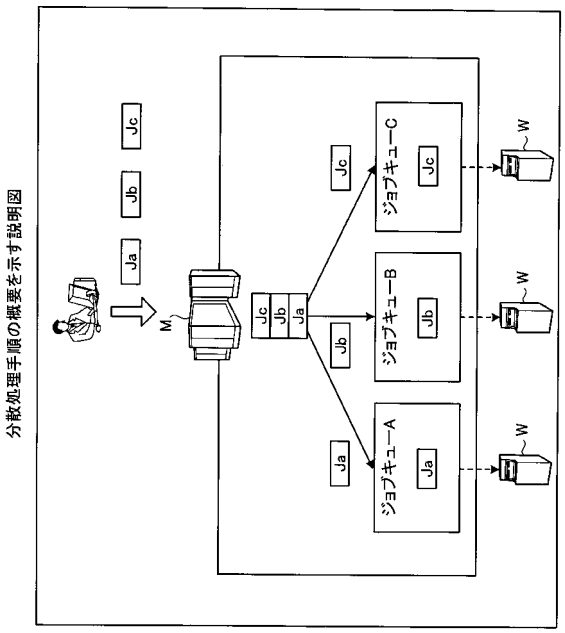
パラメータテーブルの更新処理手順の一例を示すフローチャート



【図１３】



【図１４】



【図１５】

ワーカ管理テーブルの記憶内容を示す説明図（その2）

1500

ワーカ識別子	IPアドレス	状態	実行可能タイプ
W1	xxx.yyy.z.t	使用中	A,C
W2	xxx.yyy.c.d	停止中	A,B
・・・	・・・	・・・	・・・
Wi	xxx.yyy.g.j	空き	A,B,C
・・・	・・・	・・・	・・・
Wm	xxx.yyy.k.l	使用中	A

【図１６】

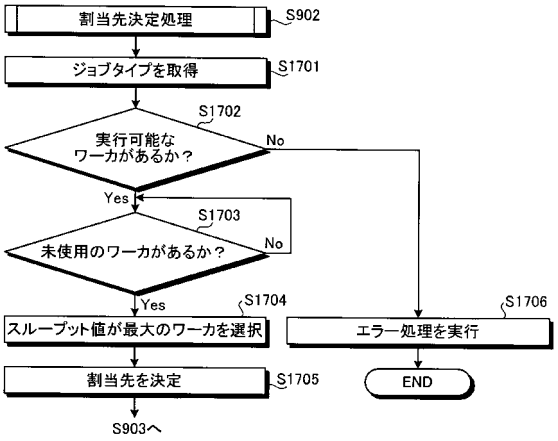
スループットテーブルの記憶内容を示す説明図（その2）

1600

ワーカ識別子	スループット値（T）		
	A	B	C
W1	T1a	—	T1c
W2	—	—	—
・・・	・・・	・・・	・・・
Wi	Tia	Tib	Tic
・・・	・・・	・・・	・・・
Wm	Tma	—	—

【図１７】

割当先決定処理手順の他の一例を示すフローチャート



フロントページの続き

(56)参考文献 特開２０００－２４２６１４（ＪＰ，Ａ）
特開２００３－２０８４１４（ＪＰ，Ａ）
特開平０４－２２３５４８（ＪＰ，Ａ）
特開平０５－０１２２２８（ＪＰ，Ａ）
特開２００９－１６９７５７（ＪＰ，Ａ）
特開２００５－１４８９１１（ＪＰ，Ａ）
特開２００４－０３８２２６（ＪＰ，Ａ）
特開２００１－３１２４８５（ＪＰ，Ａ）
米国特許第０６７５７７３０（ＵＳ，Ｂ１）
米国特許第０７０９３２５０（ＵＳ，Ｂ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ ９／４６－ ９／５４
Ｇ０６Ｆ１５／１６－１５／１７７