

(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2017-191410

(P2017-191410A)

(43) 公開日 平成29年10月19日(2017. 10. 19)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 15/173 (2006.01)	G06F 15/173 665D	5B045
G06F 13/36 (2006.01)	G06F 13/36 310E	5B061
G06F 13/38 (2006.01)	G06F 13/38 340A	5B077

審査請求 未請求 請求項の数 5 O L (全 15 頁)

(21) 出願番号	特願2016-79707 (P2016-79707)	(71) 出願人	000005223
(22) 出願日	平成28年4月12日 (2016. 4. 12)		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	110002147
			特許業務法人酒井国際特許事務所
		(72) 発明者	安島 雄一郎
			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
		Fターム(参考)	5B045 BB42 BB54
			5B061 FF05
			5B077 BA07

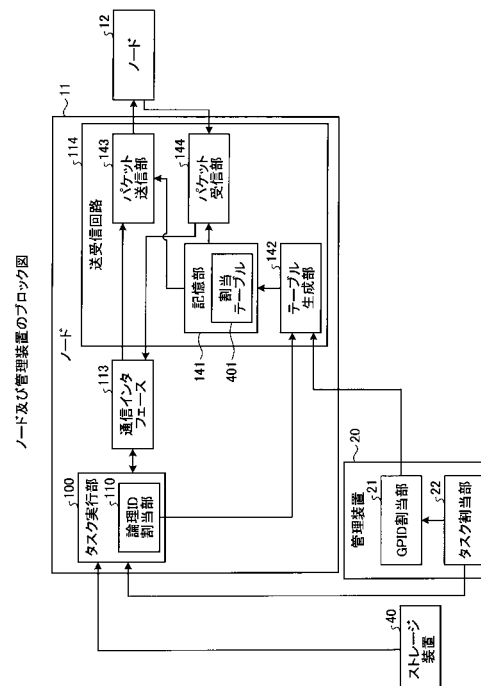
(54) 【発明の名称】 並列処理装置及び通信制御方法

(57) 【要約】

【課題】 簡易な構成で処理能力を向上させた並列処理装置及び通信制御方法を提供する。

【解決手段】 パケット送信部143は、所定処理を表す第1識別子及び宛先通信インタフェースを表す論理識別子である第2識別子を宛先として付加したパケットをRDMA通信でノード12へ送信する。また、通信インタフェース113は複数存在する。パケット受信部144は、ノード12から送信されたパケットをRDMA通信で受信し、受信したパケットに付加された第1識別子及び第2識別子を基に、受信したパケットの宛先となる所定処理で使用される通信インタフェース113を選択し、選択した通信インタフェース113に受信したパケットを転送する。

【選択図】 図2



【特許請求の範囲】**【請求項 1】**

第 1 情報処理装置及び第 2 情報処理装置を有する並列処理装置であって、
前記第 1 情報処理装置は、
所定処理を表す第 1 識別子及び宛先通信インタフェースを表す論理識別子である第 2 識別子を宛先として付加したパケットを R D M A 通信で前記第 2 情報処理装置へ送信する送信部を備え、

前記第 2 情報処理装置は、
複数の通信インタフェースと、

前記第 1 情報処理装置から送信されたパケットを R D M A 通信で受信し、受信したパケットに付加された前記第 1 識別子及び前記第 2 識別子を基に、前記受信したパケットの宛先となる前記所定処理で使用する通信インタフェースを選択し、選択した通信インタフェースに前記受信したパケットを転送する受信部とを備えた

ことを特徴とする並列処理装置。

【請求項 2】

前記第 2 情報処理装置で実行される、それぞれが異なる用途の通信を行う前記所定処理を含む複数の処理のそれぞれに前記第 1 識別子を含む一意の大域識別子を割り当てる第 1 割当部を備えた管理装置をさらに備えたことを特徴とする請求項 1 に記載の並列処理装置。

【請求項 3】

前記所定処理は、1 つ又は複数のプロセスを有する各並列計算タスクであり、

前記第 2 情報処理装置は、複数の前記通信インタフェースの中で、各前記プロセスが使用する通信インタフェースのそれぞれに異なる第 2 識別子を割り当てる第 2 割当部をさらに備えた、

ことを特徴とする請求項 1 又は 2 に記載の並列処理装置。

【請求項 4】

前記第 1 情報処理装置は、

前記所定処理を実行する第 1 実行部をさらに有し、

前記第 2 情報処理装置は、

前記所定処理を前記第 1 実行部と協働して実行する第 2 実行部をさらに有し、

前記送信部は、前記第 1 実行部により実行された所定処理によるパケット送信の場合、前記第 1 識別子及び前記第 2 識別子を宛先として付加したパケットを R D M A 通信で第 2 情報処理装置へ送信し、

前記受信部は、受信したパケットに付加された前記第 1 識別子を基に前記第 2 実行部により実行される前記所定処理を特定し、前記第 2 識別子を基に前記所定処理で使用するパケットの宛先となる前記通信インタフェースを選択し、選択した通信インタフェースに前記受信したパケットを転送する受信部とを備えた

ことを特徴とする請求項 1 ～ 3 のいずれか一つに記載の並列処理装置。

【請求項 5】

第 1 情報処理装置及び複数の通信インタフェースを有する第 2 情報処理装置の間の通信制御方法であって、

前記第 1 情報処理装置に、所定処理を表す第 1 識別子及び宛先通信インタフェースを表す論理識別子である第 2 識別子を宛先として付加したパケットを R D M A 通信で前記第 2 情報処理装置へ送信させ、

前記第 2 情報処理装置に、前記第 1 情報処理装置から送信されたパケットを R D M A 通信で受信させ、受信したパケットに付加された前記第 1 識別子及び前記第 2 識別子を基に、前記受信したパケットの宛先となる前記所定処理で使用する通信インタフェースを選択させ、選択した通信インタフェースへ前記受信したパケットを転送させる

ことを特徴とする通信制御方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、並列処理装置及び通信制御方法に関する。

【背景技術】

【0002】

近年、H P C (High Performance Computing) 分野では、それぞれがプロセッサとメモリを備え、独立してO S (Operating System) を実行するノードを、インターコネクトによって多数接続した構成を有する分散メモリ型並列計算機が主流となっている。このような分散メモリ型並列計算機では、各ノードでプロセスが起動され、起動されたプロセスは相互に通信してデータを交換しながら並列計算を行う。インターコネクトは、ノード間を接続するネットワークと、ノードがネットワークに接続するためのデバイスを有する。

10

【0003】

2つのノード間でデータを確実に転送するには、ノード間を接続するネットワーク及びデバイスの特性に合わせた手順で通信が制御される。ノード間を接続するネットワーク及びノードがネットワークに接続するためのデバイスの特性に合わせた手順は、プロトコルとも呼ばれる。

【0004】

インターネットで使用されるT C P (Transmission Control Protocol) / I P (Internet Protocol) プロトコルのプロトコル処理は、一般的にO S のプロトコルスタックが行う。各プロセスは、B a r k l e y S o c k e t などのソフトウェア・インタフェースによって、プロトコルスタックに対するデータ入出力を行う。プロトコルスタックによるプロトコル処理は、O S のシステムプロセスで実行される。すなわち、プロトコル処理を行う場合、割り込みなどでプロセッサが実行する処理がO S カーネルに移り、その間プロセッサは、他の計算を行うことが困難である。

20

【0005】

プロセッサの演算効率を上げる、すなわち実行性能を理論性能に近づけるには、プロセッサは、プロトコル処理を行わない方がよい。そこで、H P C 分野では、一般的にR D M A (Remote Direct Memory Access) と呼ばれる方式でデータを転送する。R D M A は、送信元プロセスが管理するメモリをインターコネクトのデバイスが直接読み出し、ノード間を接続するネットワークでデータを転送し、宛先プロセスが管理するメモリに直接書き込む技術である。

30

【0006】

このR D M A 方式であっても、他のプロトコル処理と同様に、通信開始制御を受けて処理が開始される。R D M A における通信開始制御は、ハードウェアの観点では、メモリ空間にマップされるメモリマップドレジスタによって通信開始を制御する方式が一般的である。また、ソフトウェアの観点では、R D M A における通信開始制御は、インターコネクトも入出力デバイスの一種であることから、O S が各プロセスの通信開始要求を取りまとめ、デバイスドライバが代表して通信を制御する方式が一般的である。デバイスドライバとは、様々なプロセスからの入出力機器への制御要求を取りまとめ、代表で制御するためのO S の仕組みである。

40

【0007】

デバイスドライバ経由の通信開始制御は、プロセスの処理を中断してO S カーネルの処理に移行する処理を伴う。この処理は、ソフトウェアの観点からはプロセススイッチと呼ばれ、ハードウェアの観点からはコンテキストスイッチと呼ばれる。プロセスの処理を中断してO S カーネルの処理に移行する処理は、オーバヘッドの大きい処理のため、プロセッサの計算時間が重要であるH P C においては、回避することが好ましい。例えば、通信インタフェースと呼ばれる制御レジスタの組を複数用意し、各プロセスの仮想メモリ空間にそのプロセスが占有利用する通信インタフェースを割り当てることで、デバイスドライバ経由の通信開始制御を回避することができる。この通信インタフェースは、インターコ

50

ネクトにおけるノードがネットワークに接続するためのデバイスにあたる。

【0008】

ここで、複数の通信インタフェースを有するインターコネクトを搭載する並列計算機では、ノード内の各プロセスだけでなく、ファイル入出力やシステム制御などのシステム向けの通信を行う処理にも通信インタフェースが割り当てられる。ここで、システム向けの通信とは、ノード動作時に経常的に動作しているプロセスにおける通信であり、例えば、システム全体を管理するプログラムによる管理用の通信や、ストレージ装置との通信などである。

【0009】

システムソフトウェアによるシステム向けの通信を行う処理に対する通信インタフェースの割り当ては、固定的である傾向がある。このような固定的な通信インタフェースの割り当てでは、通信資源の拡張は困難である。また、固定的に通信インタフェースを割り当てた場合、仮想マシンを導入する際には、ソフトウェアによるインターコネクトの仮想化が行われることになる。

【0010】

なお、クラスタシステム全体に割り当てた大域識別子と、各計算機内で用いる局所識別子との対応関係を用いた計算機クラスタの従来技術がある。

【先行技術文献】

【特許文献】

【0011】

【特許文献1】特開2003-316637号公報

【発明の概要】

【発明が解決しようとする課題】

【0012】

しかしながら、仮想化に対応しているインターコネクトでは、実際の通信インタフェースの数を隠蔽するために、OS及びデバイスドライバが通信インタフェースを管理する。その場合、システム向け通信を行う処理であっても動的に通信インタフェースが割り当てられる。

【0013】

ただし、動的に割り当てられた通信インタフェース同士で通信を行うには、ノード間で事前に、TCP (Transmission Control Protocol) / IP (Internet Protocol) などを用いて通信インタフェースの識別子の交換が行われる。そのため、実質的に他の通信手段を用いることになり、システム向け通信用途としての使用には適さない。このような理由から、システム向け通信を行う処理に対して動的に通信インタフェースを割り当てることは好ましくない。

【0014】

一方、仮想化に対応していない通信インターコネクトでは、システム向け通信用途で通信インタフェースを固定的に使用した場合、仮想マシン導入時に仮想マシン間で通信インタフェースの競合が発生するおそれがある。この場合、競合の解消のためにソフトウェアによる仮想化が用いられるが、ソフトウェア処理のためのオーバーヘッドが大きいため、HPC分野で用いることは困難である。このように、システム向け通信用途で通信インタフェースを固定的に使用した場合、他の処理に影響を与え、並列処理装置の処理能力の低下を招く。

【0015】

また、仮想マシン間の通信インタフェースの競合回避として、仮想化した入出力デバイスを多重化する方法が考えられる。例えば、従来、仮想化された複数の入出力デバイスを仮想マシン毎に割り当てることで、入出力デバイスを多重化することが行われてきた。この方式をインターコネクトに適用した場合、仮想マシン導入時の仮想マシン間での通信インタフェースの競合は避けられる。しかし、このような方法で仮想的な入出力デバイスを多重化しても、実体は1つであるので、設定レジスタの競合回避などをハードウェアで実

10

20

30

40

50

装することになり、実装が複雑になってしまう。

【0016】

なお、大域識別子と局所識別子との対応関係を用いる計算機クラスタの従来技術は、クラスタにおけるリソースの管理に用いられる情報であり、ノード間の通信に用いることは困難である。

【0017】

開示の技術は、上記に鑑みてなされたものであって、簡易な構成で処理能力を向上させた並列処理装置及び通信制御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】

【0018】

本願の開示する並列処理装置及び通信制御方法の一つの態様において、並列処理装置は第1情報処理装置及び第2情報処理装置を有する。前記第1情報処理装置は、所定処理を表す第1識別子及び宛先通信インタフェースを表す論理識別子である第2識別子を宛先として付加したパケットをRDMA通信で前記第2情報処理装置へ送信する送信部を備える。前記第2情報処理装置は、複数の通信インタフェースと、前記第1情報処理装置から送信されたパケットをRDMA通信で受信し、受信したパケットに付加された前記第1識別子及び前記第2識別子を基に、前記受信したパケットの宛先となる前記所定処理で 사용되는通信インタフェースを選択し、選択した通信インタフェースに前記受信したパケットを転送する受信部とを備える。

【発明の効果】

【0019】

本願の開示する並列処理装置及び通信制御方法の一つの態様によれば、簡易な構成で処理能力を向上させることができるという効果を奏する。

【図面の簡単な説明】

【0020】

【図1】図1は、並列処理装置の構成図である。

【図2】図2は、ノード及び管理装置のブロック図である。

【図3】図3は、割当テーブルの一例を表す図である。

【図4】図4は、GPID及び論理IDを用いたノード間のRDMA通信を説明するための図である。

【図5】図5は、GPID及び論理IDの割当処理のフローチャートである。

【図6】図6は、通信インタフェースに受信したパケットを割り当てる処理のフローチャートである。

【発明を実施するための形態】

【0021】

以下に、本願の開示する並列処理装置及び通信制御方法の実施例を図面に基づいて詳細に説明する。なお、以下の実施例により本願の開示する並列処理装置及び通信制御方法が限定されるものではない。

【実施例】

【0022】

図1は、並列処理装置の構成図である。図1に示すように、本実施例に係る並列処理装置1は、ノード11～13を含む複数のノードを有する。ノード11～13を含む各ノードを区別しない場合、「ノード10」という。ノード10は、第1情報処理装置及び第2情報処理装置の一例にあたる。また、並列処理装置1は、管理装置20、ネットワーク30及びストレージ装置40を有する。

【0023】

ネットワーク30は、インターコネクトを形成するネットワークであり、ノード10同士を接続する。

【0024】

管理装置20は、各ノード10にTCP/IPなどを用いて接続される。管理装置20

10

20

30

40

50

は、各ノード 10 への並列計算タスクの割り当てや各ノード 10 の異常の検出などを行う。

【0025】

ストレージ装置 40 は、各ノード 10 に TCP/IP などを用いて接続される。ストレージ装置 40 は、並列計算に用いるプログラムやデータを格納する。ここで、本実施例では、並列処理装置 1 にノード 10 とは別にプログラムやデータを格納するストレージ装置 40 を配置したが、各ノード 10 にハードディスクを搭載し、そのハードディスクの中にプログラムやデータを格納する構成にしてもよい。

【0026】

ノード 11 は、プロセッサ 111、メモリ 112、通信インタフェース 113 及び送受信回路 114 を有する。ノード 12 は、プロセッサ 121、メモリ 122、通信インタフェース 123 及び送受信回路 124 を有する。ノード 13 は、プロセッサ 131、メモリ 132、通信インタフェース 133 及び送受信回路 134 を有する。ノード 11、12 及び 13 は、いずれも同じ動作をするので、ここでは、ノード 11 を例に説明する。

【0027】

プロセッサ 111 は、演算処理部である。プロセッサ 111 は、OS を動作させる。さらに、プロセッサ 111 は、ストレージ装置 40 に格納されたプログラムを読み出し、メモリ 112 上にプロセスを展開して実行する。そして、プロセッサ 111 は、プロセスを実行することで、メモリ 112 を用いて演算処理を行う。また、プロセッサ 111 は、通信インタフェース 113 を用いて他のノード 12 や 13 との間で通信を行う。また、プロセッサ 111 は、通信インタフェース 113 に対して RDMA による通信（以下では、「RDMA 通信」という。）を指示することで、通信インタフェース 113 に RDMA 通信を実行させる。

【0028】

メモリ 112 は、主記憶装置である。メモリ 112 は、例えば、RAM (Random Access Memory) である。

【0029】

通信インタフェース 113 は、インターコネクトにおけるノード 11 がネットワーク 30 に接続するためのデバイスである。通信インタフェース 113 は、それぞれ複数の物理的な通信インタフェースを有する。ここで、通信インタフェース 113、123 及び 133 といった各ノード 10 に含まれる通信インタフェースの数に特に制限は無く、ノード 10 毎に通信インタフェースの数が異なってもよい。

【0030】

通信インタフェース 113 は、RDMA によりデータ送信を行うことができる。通信インタフェース 113 は、プロセッサ 111 から独立して動作する専用回路などで構成される演算処理装置を有する。通信インタフェース 113 による RDMA 通信を用いたデータ送信及び受信応答などの制御は、この演算処理装置により実現される。通信インタフェース 113 は、それぞれに物理的な識別子である物理 ID (Identifier) が割り当てられる。また、通信インタフェース 113 は、それぞれに後述する論理 ID が割り当てられる。また、通信インタフェース 113 は、プロセッサ 111 が実行するプロセスにより、各プロセスが使用する通信インタフェースとして対応付けられる。

【0031】

通信インタフェース 113 は、RDMA 通信でデータを他のノード 10 へ送信する場合、宛先の通信インタフェースの物理 ID が付加されたパケットを送受信回路 114 に送信する。また、RDMA 通信でデータを他のノード 10 から受信する場合、通信インタフェース 113 は、送受信回路 114 からパケットを受信し、プロセッサ 111 によって実行される対応するプロセスにパケットを転送する。

【0032】

送受信回路 114 は、通信インタフェース 113 が実行する RDMA 通信を仲介する。具体的には、送受信回路 114 は、通信インタフェース 113 が送出したパケットを受信

10

20

30

40

50

する。そして、送受信回路 114 は、宛先を後述する G P I D (Group Identifier) 及び論理 I D に変更する。そして、送受信回路 114 は、宛先を変更したパケットをネットワーク 30 を介して他のノード 10 へ送信する。

【0033】

また、送受信回路 114 は、他のノード 10 からネットワーク 30 を介して受信したパケットの宛先から使用する通信インタフェース 113 及び宛先となるプロセスを特定する。そして、送受信回路 114 は、特定した通信インタフェース 113 を介して特定したプロセスにパケットを送信する。

【0034】

ここで、本実施例に係る並列処理装置 1 において、ノード 11 のメモリ 112 に格納されたデータをノード 12 との間で R D M A により送受信する処理について詳細に説明する。

10

【0035】

図 2 は、ノード及び管理装置のブロック図である。ここでは、ノード 11 を含むノード 10 に並列計算タスクを実行させる場合で説明する。並列計算タスクの実行は、複数のノード 10 において 1 つ又は複数のプロセスがタスク実行部 100 により実行されることで実現される。すなわち、並列計算タスクは、タスク実行部 100 で実行されるプロセスを含む各ノード 10 で実行される複数のプロセスにより実現される。以下では、ノード 11 を含むノード 10 に実行させる特定の並列計算タスクを、「実行タスク」という。また、並列計算タスクを実現するための各プロセスを、「その並列計算タスクに属するプロセス」という。

20

【0036】

管理装置 20 は、G P I D 割当部 21 及びタスク割当部 22 を有する。タスク割当部 22 は、実行する並列計算タスク及びその並列計算タスクで使用するノード数などの資源の情報の入力を操作者から受ける。そして、タスク割当部 22 は、指定された並列計算タスクを実行させるノード 10、言い換えれば、並列計算タスクを割り当てるノード 10 を決定する。ここでは、タスク割当部 22 は、ノード 11 を含む複数のノード 10 に実行タスクを割り当てる。

【0037】

タスク割当部 22 は、実行タスクを実行するためにノード 11 に実行させるプログラムの情報をノード 11 のタスク実行部 100 に通知する。さらに、タスク割当部 22 は、実行タスクを割り当てたノード 11 を含むノード 10 の情報を G P I D 割当部 21 へ通知する。

30

【0038】

G P I D 割当部 21 は、並列処理装置 1 で実行される並列計算タスク毎に並列計算タスクの識別子である G P I D を割り当てる。具体的には、G P I D 割当部 21 は、並列計算タスクのノード 10 に対する割り当てがタスク割当部 22 から通知される毎に、数字を 1 つインクリメントするなどして、それぞれの並列計算タスクに一意的な G P I D を割り当てる。例えば、G P I D 割当部 21 は、実行タスクを割り当てたノード 11 を含むノード 10 の情報をタスク割当部 22 から取得する。そして、G P I D 割当部 21 は、実行タスクに一意的な G P I D を割り当てる。この G P I D が、「第 1 識別子」及び「大域識別子」の一例にあたる。

40

【0039】

さらに、G P I D 割当部 21 は、並列計算タスクが割り当てられた各ノード 10 にその割り当てられた並列計算タスクに対応する G P I D を通知する。例えば、G P I D 割当部 21 は、実行タスクに対応する G P I D をノード 11 の送受信回路 114 のテーブル生成部 142 に通知する。この G P I D 割当部 21 は、「第 1 割当部」の一例にあたる。

【0040】

ここで、本実施例では、並列計算タスクに対する G P I D の割り当てを例に説明したが、G P I D の割り当て対象はこれに限らず、G P I D 割当部 21 は、例えば、並列処理装

50

置 1 が実行する特定の通信を行う処理毎に G P I D を割り当てる。特定の通信を行う処理毎とは、例えば、ストレージとの通信、システム全体を管理するプログラムによる管理用の通信を行う処理といった、並列計算処理装置において決められた通信を行う処理毎である。この並列計算タスクを含む特定の通信を行う処理及び並列計算タスクを実行する仮想マシンが、「所定処理」の一例にあたる。ここで、仮想マシンとは、コンピュータの動作をエミュレートするソフトウェアやフレームワークであり、処理の一例として考えられる。

【0041】

ノード 11 は、タスク実行部 100、通信インタフェース 113 及び送受信回路 114 を有する。図 2 では、1 つの通信インタフェース 113 を図示しているが、実際には、図 1 で図示したように、通信インタフェース 113 は、複数の通信インタフェースを有する。以下では、通信インタフェース 113 は、ノード 11 に搭載された複数の通信インタフェースの何れかであるとする。

10

【0042】

タスク実行部 100 は、図 2 のプロセッサ 111 及びメモリ 122 により実現される。タスク実行部 100 は、論理 I D 割当部 110 を有する。

【0043】

タスク実行部 100 は、実行タスクを実行するためのノード 11 で実行するプログラムの情報をタスク割当部 22 から取得する。そして、タスク実行部 100 は、指定されたプログラムをストレージ装置 40 から取得する。その後、タスク実行部 100 は、取得したプログラムを実行してプロセスを生成する。

20

【0044】

プロセス生成後、タスク実行部 100 は、自己が実行する実行タスクに属するプロセス及び実行タスクが割り当てられた他のノード 10 で生成された実行タスクに属するプロセスの全てと同期をとる。タスク実行部 100 は、この同期のための通信を、例えば、適当な通信インタフェース 113 及び送受信回路 114 を介して他のノード 10 で行う。

【0045】

また、タスク実行部 100 の論理 I D 割当部 110 は、タスク実行部 100 がプログラムを実行することで、そのプログラムから生成されるプロセスが使用する通信インタフェース 113 の個数を取得する。そして、論理 I D 割当部 110 は、取得した個数に合わせてプロセスが使用する通信インタフェース 113 を決定する。さらに、論理 I D 割当部 110 は、プロセスが使用する通信インタフェース 113 のそれぞれに実行タスクを実行するノード 10 の全てに亘って一意の論理 I D を割り当てる。

30

【0046】

例えば、論理 I D 割当部 110 は、異なるプロセスであっても同じ並列計算タスクに含まれるプロセスであれば、各プロセスが使用する通信インタフェース 113 に対して、重複しないように論理 I D を割り当てる。これに対して、異なる並列計算タスクに含まれるプロセスであれば、論理 I D 割当部 110 は、それぞれのプロセス間での重複を許して通信インタフェース 113 に論理 I D を割り当てる。この論理 I D が、「第 2 識別子」の一例にあたる。

40

【0047】

論理 I D 割当部 110 は、実行タスクの情報とともに、その実行タスクのプロセスが使用する通信インタフェース 113 に割り当てた論理 I D とその通信インタフェース 113 の物理 I D との対応関係をテーブル生成部 142 へ通知する。この論理 I D 割当部 110 は、「第 2 割当部」の一例にあたる。

【0048】

そして、タスク実行部 100 は、論理 I D 割当部 110 による論理 I D の割り当て後、自己が実行する実行タスクに属するプロセス及び実行タスクが割り当てられた他のノード 10 で生成された実行タスクに属するプロセスの全てと同期をとる。

【0049】

50

その後、タスク実行部 100 は、生成したプロセスを実行し、他のノード 10 が実行するプロセスと協働させることで、実行タスクを実行する。このプロセスの実行により、タスク実行部 100 は、他のノード 10 との R D M A 通信の実行を決定し、R D M A 通信を行うプロセスに対応する他のノード 10 が実行するプロセスの通信インタフェース 113 との間で R D M A 通信を行わせる。例えば、ノード 11 とノード 12 との間における R D M A 通信について説明する。その場合、タスク実行部 100 は、R D M A 通信を行うプロセスがノード 12 との通信で使用する通信インタフェース 113 に、ノード 12 との間での R D M A 通信によるデータの送信又は受信を指示する。このタスク実行部 100 が、「第 1 実行部」及び「第 2 実行部」の一例にあたる。

【0050】

通信インタフェース 113 は、タスク実行部 100 から R D M A 通信の通知を受ける。ここで、通信インタフェース 113 は、R D M A 通信を行うノード 12 の通信インタフェース 123 の論理 I D の情報を予め有する。通信インタフェース 113 は、R D M A 通信として、例えば、ノード 12 に対する P u t 又は G e t の要求を受ける。通信インタフェース 113 は、P u t 要求を受けた場合、論理 I D を用いてデータを格納したパケットをノード 12 の通信インタフェース 123 へ送信する。また、通信インタフェース 113 は、G e t 要求を受けた場合、論理 I D を用いてデータの送信要求のパケットをノード 12 の通信インタフェース 123 へ送信する。その後、通信インタフェース 113 は、データを格納したパケットを送受信回路 114 を介してノード 12 の通信インタフェース 123 から受信する。そして、通信インタフェース 113 は、受信したパケットに格納されたデータを、パケットで指定されたメモリ 112 の領域へ格納する。

【0051】

通信インタフェース 113 は、送信するパケットを生成する場合に以下のような処理を行う。まず、通信インタフェース 113 は、対応するプロセスが属する実行タスクの G P I D 及び R D M A 通信の予め決められた通信先であるノード 12 の通信インタフェース 123 の論理 I D をパケットの宛先としてパケットのヘッダに格納する。さらに、通信インタフェース 113 は、パケットの送信元として自己の物理 I D をパケットのヘッダに格納する。そして、通信インタフェース 113 は、生成したパケットを送受信回路 114 のパケット送信部 143 へ出力する。

【0052】

また、パケットを受信する場合、通信インタフェース 113 は、R D M A 通信の予め決められた通信先であるノード 12 の通信インタフェース 123 から送信されたパケットを後述する送受信回路 114 のパケット受信部 144 から受信する。

【0053】

送受信回路 114 は、記憶部 141、テーブル生成部 142、パケット送信部 143 及びパケット受信部 144 を有する。記憶部 141、テーブル生成部 142、パケット送信部 143 及びパケット受信部 144 は、例えば、送受信回路 114 を構成する専用回路により実現される。

【0054】

記憶部 141 は、データを保持するデータ保持回路である。以下では、記憶部 141 が、割当テーブル 401 を保持するとして説明するが、実際には、記憶部 141 は、情報の対応関係を記憶する。

【0055】

テーブル生成部 142 は、タスク実行部 100 が実行する実行タスクに割り当てられた G P I D を G P I D 割当部 21 から受信する。さらに、テーブル生成部 142 は、実行タスクの情報とともに、その実行タスクのプロセスが使用する通信インタフェース 113 に割り当てられた論理 I D とその通信インタフェース 113 の物理 I D との対応関係を論理 I D 割当部 110 から取得する。

【0056】

次に、テーブル生成部 142 は、各通信インタフェース 113 を使用するプロセスが属

10

20

30

40

50

する実行タスクに割り当てられた G P I D を取得する。そして、テーブル生成部 1 4 2 は、各通信インタフェース 1 1 3 の物理アドレスと、取得した G P I D 及びその通信インタフェース 1 1 3 に割り当てられた論理 I D との対応を表す図 3 に示すような割当テーブル 4 0 1 を作成する。図 3 は、割当テーブルの一例を表す図である。その後、テーブル生成部 1 4 2 は、生成した割当テーブル 4 0 1 を記憶部 1 4 1 に記憶させる。

【0057】

図 3 に示すように、割当テーブル 4 0 1 には、通信インタフェース 1 1 3 の物理 I D に対応させて、G P I D 及び論理 I D が登録される。ここで、割当テーブル 4 0 1 では、ノード 1 1 の通信インタフェース 1 1 3 の物理 I D と、G P I D 及び論理 I D の組とが一意に対応する。したがって、割当テーブル 4 0 1 を用いることで、G P I D 及び論理 I D の組から通信インタフェース 1 1 3 の物理 I D が特定できる。

10

【0058】

パケット送信部 1 4 3 は、R D M A 通信のパケット送信時に通信インタフェース 1 1 3 からパケットの入力を受ける。そして、パケット送信部 1 4 3 は、R D M A 通信でパケットをノード 1 2 などの他のノード 1 0 へ送信する。このパケット送信部 1 4 3 が、「送信部」の一例にあたる。

【0059】

パケット受信部 1 4 4 は、R D M A 通信でパケットをノード 1 2 などの他のノード 1 0 から受信する。そして、パケット受信部 1 4 4 は、受信したパケットのヘッダに格納されたパケットの宛先を表す G P I D 及び論理 I D の組を取得する。次に、パケット受信部 1 4 4 は、ノード 1 1 における G P I D 及び論理 I D の組に対応する通信インタフェース 1 1 3 の物理 I D を割当テーブル 4 0 1 から取得する。そして、パケット受信部 1 4 4 は、取得した物理 I D を有する通信インタフェース 1 1 3 を選択する。その後、パケット受信部 1 4 4 は、受信したパケットを選択した通信インタフェース 1 1 3 へ送信する。このパケット受信部 1 4 4 は、「受信部」の一例にあたる。

20

【0060】

このように、パケット受信部 1 4 4 は、通信インタフェース 1 1 3 に割り当てられた論理 I D と実行タスクの G P I D を用いて、そのパケットの宛先のプロセスが使用する通信インタフェース 1 1 3 にパケットを送信することができる。すなわち、各ノード 1 0 は、通信インタフェース 1 1 3 に動的に割り当てられた論理 I D を用いて R D M A 通信を確実にすることができる。

30

【0061】

次に、図 4 を参照して、ノード 1 1 とノード 1 2 との間の R D M A 通信についてまとめて説明する。図 4 は、G P I D 及び論理 I D を用いたノード間の R D M A 通信を説明するための図である。図 4 では、説明の都合上、ソフトウェアとハードウェアとを混在させて記載している。

【0062】

ここでは、ノード 1 1 及び 1 2 において並列計算タスク 1 0 1 及び 1 0 2 が実行されている。そして、ノード 1 1 には通信インタフェース 1 1 3 として通信インタフェース 3 1 1 ~ 3 1 4 が搭載されており、ノード 1 2 には通信インタフェース 1 1 3 として通信インタフェース 3 2 1 ~ 3 2 4 が搭載されている場合で説明する。

40

【0063】

ノード 1 1 の通信インタフェース 3 1 1 ~ 3 1 4 は、物理 I D としてそれぞれ I F # 0 ~ # 3 を有する。そして、並列計算タスク 1 0 1 が使用する通信インタフェース 1 1 3 として、通信インタフェース 3 1 1 ~ 3 1 3 が割り当てられる。また、並列計算タスク 1 0 1 に、G P I D # 0 が割り当てられる。そして、通信インタフェース 3 1 1 ~ 3 1 3 に、論理 I D として、L I F # 0 ~ # 2 が割り当てられる。また、並列計算タスク 1 0 2 が使用する通信インタフェース 1 1 3 として、通信インタフェース 3 1 4 が割り当てられる。また、並列計算タスク 1 0 2 に、G P I D # 1 が割り当てられる。そして、通信インタフェース 3 1 4 に、論理 I D として、L I F # 0 ~ # 2 が割り当てられる。

50

【0064】

ノード12の通信インタフェース321～324は、物理IDとしてそれぞれIF#0～#4を有する。ここで、ノード10間では、通信インタフェース113の物理IDは重なってもよい。そして、並列計算タスク101が使用する通信インタフェース113として、通信インタフェース321及び322が割り当てられる。そして、通信インタフェース321及び322に、論理IDとして、それぞれLIF#0及び#1が割り当てられる。また、並列計算タスク102が使用する通信インタフェース113として、通信インタフェース323及び324が割り当てられる。そして、通信インタフェース323及び324に、論理IDとして、それぞれLIF#0及び#1が割り当てられる。

【0065】

また、ノード11の送受信回路114は、割当テーブル401が格納されている。また、ノード12の送受信回路124は、割当テーブル402が格納されている。そして、割当テーブル401及び402には、ノード11に搭載された通信インタフェース311～314の物理IDであるIF#0～4に対応させて、GPID及び論理IDが登録されている。また、割当テーブル401及び402には、ノード12に搭載された通信インタフェース321～324の物理IDであるIF#0～4に対応させて、GPID及び論理IDが登録されている。例えば、ここにおける割当テーブル401及び402におけるノード12に対するテーブルは、図3で示される割当テーブル401と同様である。

【0066】

ノード11の並列計算タスク101が、ノード12の並列計算タスク101とRDMA通信を行う場合について説明する。並列計算タスク101が通信インタフェース312にRDMA通信の開始を指示する。通信インタフェース312は、ノード12の論理IDがLID#1を有する通信インタフェース113と通信を行うことが予め決められている。そこで、通信インタフェース312は、ノード12の識別情報、並列計算タスク101のGPIDであるGPID#0及び論理IDとしてLID#1を宛先としてパケットのヘッダに格納する。図3では、通信インタフェース312によりヘッダに格納された宛先を宛先411として表す。そして、通信インタフェース312は、生成したパケットを送受信回路114へ送信する。送受信回路114は、宛先411に示すノード12、GPID#0及びLIF#1を宛先とするパケットをネットワーク30を介してノード12へ送信する。

【0067】

送受信回路124は、宛先411に示すノード12、GPID#0及びLIF#1を宛先とするパケットを受信する。そして、送受信回路124は、受信したパケットのヘッダからGPID#0及びLIF#1を取得する。次に、送受信回路124は、割当テーブル402の中の列421で表される対応関係から、GPID#0及びLIF#1に対応する物理IDであるIF#1を取得する。そして、送受信回路124は、物理IDがIF#1を有する通信インタフェース322へパケットを送信する。これにより、ノード11の並列計算タスク101は、ノード12の並列計算タスク101と通信を行うことができる。

【0068】

次に、図5を参照して、GPID及び論理IDの割当処理の流れについて説明する。図5は、GPID及び論理IDの割当処理のフローチャートである。

【0069】

管理装置20のタスク割当部22は、並列計算タスクを割り当てるノード10を決定する。GPID割当部21は、タスク割当部22により割り当てが行われた並列計算タスクに割り当てる新規GPIDを生成する（ステップS101）。また、タスク割当部22は、並列計算タスクを割り当てたノード10に並列計算タスクの割り当てを通知する。そして、並列計算タスクの割り当てを通知されたノード10のタスク実行部100は、並列計算タスクのプロセスを生成する。

【0070】

次に、タスク割当部22は、生成したGPIDに対応する並列計算タスクの各プロセス

10

20

30

40

50

を実行するタスク実行部 100 にその G P I D を送信することで、その並列計算タスクの全プロセスに G P I D を通知する（ステップ S 102）。

【0071】

タスク実行部 100 は、実行する並列計算タスクの全プロセス及びその並列計算タスクを実行する他のノード 10 の全プロセスの間で同期を行う（ステップ S 103）。

【0072】

論理 I D 割当部 110 は、各プロセスが使用する通信インタフェース 113 に論理 I D を割り当てる（ステップ S 104）。この時、タスク実行部 100 は、異なる並列計算タスクのプロセス間では、重複を許して論理 I D を割り当てる。

【0073】

次に、論理 I D 割当部 110 は、並列計算タスクの各プロセスが使用する通信インタフェース 113 の物理 I D、各並列計算タスクの G P I D 及び各通信インタフェース 113 の論理 I D を送受信回路 114 のテーブル生成部 142 に送信する。テーブル生成部 142 は、受信した物理 I D、G P I D 及び論理 I D から割当テーブル 401 を作成する。そして、テーブル生成部 142 は、生成した割当テーブル 401 を記憶部 141 に格納する（ステップ S 105）。

【0074】

その後、タスク実行部 100 は、実行する並列計算タスクの全プロセス及びその並列計算タスクを実行する他のノード 10 の全プロセスの間で同期を行う（ステップ S 106）。

【0075】

そして、タスク実行部 100 は、プロセスを実行し、並列計算を開始する（ステップ S 107）。

【0076】

次に、図 6 を参照して、通信インタフェース 113 に受信したパケットを割り当てる処理の流れについて説明する。図 6 は、通信インタフェースに受信したパケットを割り当てる処理のフローチャートである。

【0077】

送受信回路 114 のパケット受信部 144 は、他のノード 10 から R D M A 通信によりパケットを受信する（ステップ S 201）。

【0078】

そして、パケット受信部 144 は、受信したパケットのヘッダからパケットの宛先である並列計算タスクの G P I D 及び通信インタフェース 113 の論理 I D を取得する（ステップ S 202）。

【0079】

次に、パケット受信部 144 は、送信元のノード 10 に対応する割当テーブル 401 に登録された通信インタフェース 113 の G P I D 及び論理 I D とパケットの宛先の G P I D 及び論理 I D とを比較する（ステップ S 202）。

【0080】

そして、パケット受信部 144 は、パケットの宛先と G P I D 及び論理 I D が一致する通信インタフェース 113 が割当テーブル 401 に登録されているか否かを判定する（ステップ S 203）。

【0081】

パケットの宛先と G P I D 及び論理 I D が一致する通信インタフェース 113 が割当テーブル 401 に登録されている場合（ステップ S 203：肯定）、パケット受信部 144 は、一致した通信インタフェース 113 の物理 I D を取得する（ステップ S 204）。

【0082】

そして、パケット受信部 144 は、取得した物理 I D を有する通信インタフェース 113 にパケットを送信する（ステップ S 205）。

【0083】

これに対して、パケットの宛先と G P I D 及び論理 I D が一致する通信インタフェース

10

20

30

40

50

113が割当テーブル401に登録されていない場合（ステップS203：否定）、パケット受信部144は、受信したパケットを破棄する（ステップS206）。

【0084】

ここで、以上の説明では、並列計算タスクにGPIDを動的に割り当て、系列計算タスクに属する各プロセスが使用する通信インタフェース113に論理IDを動的に割り当てる場合で説明した。ただし、GPIDを割り当てる対象は並列計算タスクに限らない。例えば、特定の通信を行う処理、用途が予め規定された通信を行う処理についてGPIDを割り当て、その処理で使用される通信インタフェース113に論理IDを割り当ててもよい。その場合、論理IDは固定的に割り当てられてもよい。特定の通信を行う処理とは、例えば、特定の計算処理、特定のIO処理及びシステム向けの通信を行う処理などである。

10

【0085】

以上に説明したように、本実施例に係る並列処理装置は、並列計算タスク及び通信用途毎にそれぞれを識別するための大域識別子としてGPIDを割り当て、各並列計算タスク及び通信用途で使用される通信インタフェースに論理識別子を割り当てる。そして、本実施例に係る並列処理装置は、大域識別子及び論理識別子を用いてノード間で通信を行う。

【0086】

通信インタフェースに割り当てる論理識別子は、固定的に使用されても、他の並列計算タスクや通信用途の通信には影響を与えない。したがって、通信インタフェースに固定的に割り当てられた論理識別子を用いても、並列処理装置の処理能力の低下を軽減することができる。すなわち、通信に用いる識別子の割当の自由度を向上させることができる。

20

【0087】

また、本実施例に係る並列処理装置における通信では、動的に論理識別子を通信インタフェースに割り当てるが、ノード間での事前の識別子の交換を行わないため識別子交換用の通信手段を使用せずに通信を行うことができ、容易な処理で通信を行うことができる。

【0088】

また、本実施例に係る並列処理装置における大域識別子及び論理識別子を用いた通信は、ソフトウェアによる通信インタフェースの仮想化の場合のソフトウェア処理によるオーバーヘッドが発生しない。そのため、並列処理装置の処理能力の低下を軽減することができる。

30

【0089】

また、仮想マシンの導入時には、仮想マシン毎に異なる大域識別子を割り当てることで、競合を解消することができ、設定レジスタの競合回避のためのハードウェアを実装しなくても良くなる。したがって、簡易な構成で仮想マシンの競合を解消することができる。

【0090】

さらに、本実施例に係る並列処理装置は、科学技術計算用分散メモリ型並列計算機に適用することができる。科学技術計算用分散メモリ型並列計算機は、従来の数値シミュレーションに加え、大規模グラフ処理、機械学習、神経形態学的計算などの新しいアプリケーションを用いた処理での活用が期待されている。これらのアプリケーションでは、通信遅延がボトルネックとなり待ち時間が多いので、各ノードで同時に複数の並列計算タスクを実行することでプロセッサの利用率を向上させ、システム全体における並列計算タスク処理のスループットを向上することができる。そして、本実施例に係る並列処理装置は、各ノードにおいて複数の並列計算タスクの同時実行を容易にするので、科学技術計算用分散メモリ型並列計算機における新しいアプリケーションの利用を促進することができる。

40

【符号の説明】

【0091】

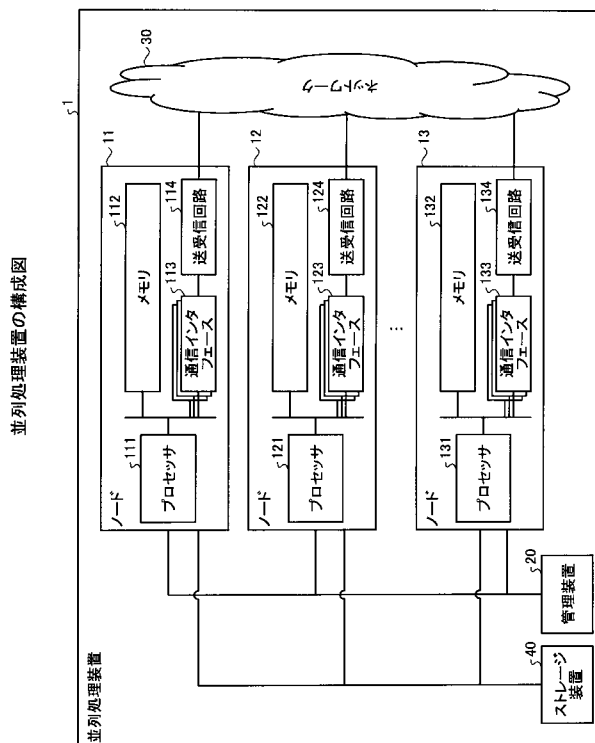
- 1 並列処理装置
- 10～13 ノード
- 20 管理装置
- 21 GPID割当部

50

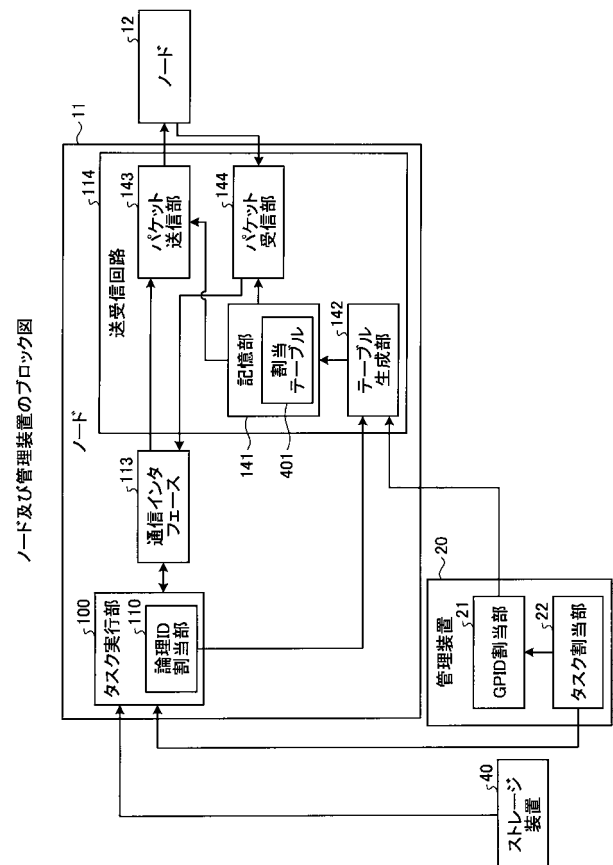
- 2 2 タスク割当部
- 3 0 ネットワーク
- 4 0 ストレージ装置
- 1 0 0 タスク実行部
- 1 0 1, 1 0 2 並列計算タスク
- 1 1 0 論理ID割当部
- 1 1 1, 1 2 1, 1 3 1 プロセッサ
- 1 1 2, 1 2 2, 1 3 2 メモリ
- 1 1 3, 1 2 3, 1 3 3 通信インタフェース
- 1 1 4, 1 2 4, 1 3 4 送受信回路
- 1 4 1 記憶部
- 1 4 2 テーブル生成部
- 1 4 3 パケット送信部
- 1 4 4 パケット受信部
- 3 1 1 ~ 3 1 4, 3 2 1 ~ 3 2 4 通信インタフェース
- 4 0 1, 4 0 2 割当テーブル

10

【図 1】



【図 2】

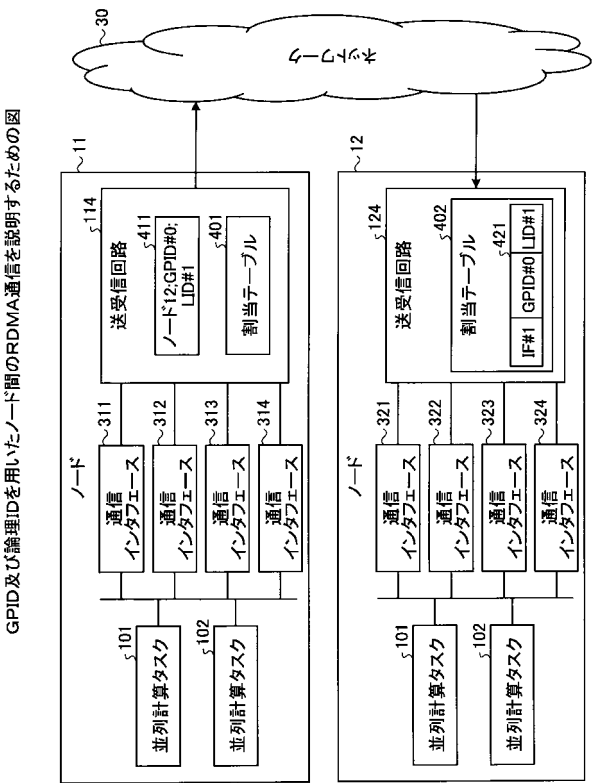


【図 3】

割当テーブルの一例を表す図

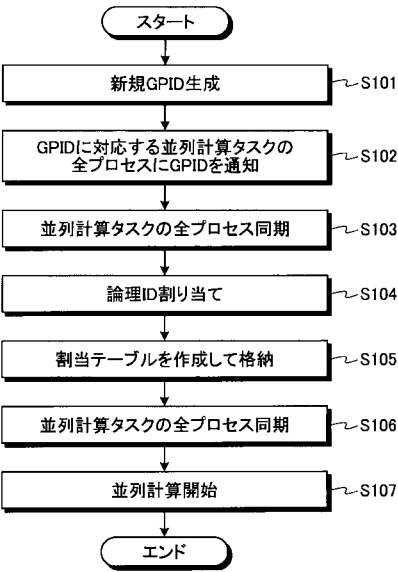
401		
通信インタフェース物理ID	GPID	論理ID
IF#0	GPID#0	LIF#0
IF#1	GPID#0	LIF#1
IF#2	GPID#1	LIF#0
IF#3	GPID#1	LIF#1

【図 4】



【図 5】

GPID及び論理IDの割当処理のフローチャート



【図 6】

通信インタフェースに受信したパケットを割り当てる処理のフローチャート

