

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第4100256号  
(P4100256)

(45) 発行日 平成20年6月11日(2008.6.11)

(24) 登録日 平成20年3月28日(2008.3.28)

(51) Int.Cl.

F 1

<b>G06F 15/17</b>	<b>(2006.01)</b>	G06F 15/17	620H
<b>G06F 3/06</b>	<b>(2006.01)</b>	G06F 3/06	301S
<b>G06F 12/08</b>	<b>(2006.01)</b>	G06F 12/08	513
<b>G06F 12/10</b>	<b>(2006.01)</b>	G06F 12/08	523E
		G06F 12/08	551J

請求項の数 5 (全 10 頁) 最終頁に続く

(21) 出願番号

特願2003-151790 (P2003-151790)

(22) 出願日

平成15年5月29日 (2003.5.29)

(65) 公開番号

特開2004-355307 (P2004-355307A)

(43) 公開日

平成16年12月16日 (2004.12.16)

審査請求日 平成17年11月8日 (2005.11.8)

前置審査

(73) 特許権者 000005108

株式会社日立製作所

東京都千代田区丸の内一丁目6番6号

(74) 代理人 100100310

弁理士 井上 学

(72) 発明者 坂口 明彦

東京都国分寺市東恋ヶ窪一丁目280番地  
株式会社日立製作所中央研究所内

審査官 久保 正典

最終頁に続く

(54) 【発明の名称】 通信方法および情報処理装置

## (57) 【特許請求の範囲】

## 【請求項 1】

I/O命令を発行する少なくとも二つ以上のCPUと、  
 該CPUから直接的または間接的にアクセス可能なディスク装置と、  
 該ディスク装置のデータの少なくとも一部のコピーを格納する前記ディスク装置に接続  
 されたディスクキャッシュと、  
 前記I/O命令を受けて該ディスク装置および該ディスクキャッシュを制御するための前  
 記CPUに接続されたI/Oプロセッサとを有し、  
 前記CPUは前記I/Oプロセッサに対して前記ディスクキャッシュ一部領域の割当要求を行う  
 機能を有し、

前記I/Oプロセッサは前記割当要求を受けると前記ディスクキャッシュの一部の領域を  
 前記CPUのメモリ空間であって当該ディスク装置とデータの入出力しない通信バッファと  
 して割り当てる機能を有し、

任意のCPUが前記通信バッファにデータを書き込み、前記任意のCPUとは異なるCPUが当  
 該データを読み込む通信手段を有し、

前記通信手段は、

前記通信バッファにこれから転送されるデータが格納されているかどうかを判定する手  
 段と、

当該通信バッファにこれから転送されるデータが格納されていなければ任意のCPUから  
 転送データを通信バッファに転送して格納する手段と、

10

20

当該通信バッファにこれから転送されるデータが格納されていれば任意のCPUが当該通信バッファから転送データを読み込み、当該バッファをクリアする手段とを含み、前記通信バッファは内部が複数のメッセージキューに分かれており、該メッセージキューを先頭から順にラウンドロビンに使用し、前回使用したキューの位置を記憶しておくことにより、通信時に確認すべきメッセージキューの位置を一意に定めることを特徴とする情報記憶装置。

#### 【請求項 2】

前記I/Oプロセッサは、前記CPUからの要求を契機に前記ディスクキャッシュの一部を前記通信バッファとして該CPUのアドレス空間に割り当てる手段を有し、該CPUは該CPUのアドレス空間にアクセスすることによってディスクキャッシュにアクセスすることを特徴とする請求項1記載の情報処理装置。

10

#### 【請求項 3】

前記I/Oプロセッサは、前記CPUからの要求を契機に前記ディスクキャッシュの一部を前記通信バッファとして該CPUのアドレス空間に割り当てる手段と、

割り当てたディスクキャッシュのアドレスとサイズを前記CPUに通知する手段を有し、

前記CPUは前記アドレスとサイズを指定して通信データを転送する請求項1記載の情報処理装置。

#### 【請求項 4】

前記CPUから直接ディスクキャッシュにアクセスするのではなく、前記転送データの格納されるメモリのアドレスをI/Oプロセッサに通知し、I/Oプロセッサが前記メモリにアクセスして前記ディスクキャッシュに割り当たられた前記通信バッファにデータを書き込むことを特徴とする請求項1記載の情報処理装置。

20

#### 【請求項 5】

前記CPUは、送信しようとする転送データの格納される送信メモリのアドレスをI/Oプロセッサに通知する手段と、転送データを受信しようとする受信メモリのアドレスをI/Oプロセッサに通知する手段と、を有し、

前記I/Oプロセッサは、前記アドレスに従い前記送信メモリから転送データをディスクキャッシュ上の通信バッファに転送する手段と、前記アドレスに従い転送データをディスクキャッシュ上の通信バッファから前記受信メモリに転送する手段と、を有する請求項4記載の情報処理装置。

30

#### 【発明の詳細な説明】

##### 【0001】

##### 【発明の属する技術分野】

本発明は、高速でセキュアな通信を行うために、ディスク装置内部に持つディスクキャッシュを介して、複数CPU間でデータの授受を行うことを特徴とした通信方法に関する

##### 【0002】

##### 【従来の技術】

##### 【特許文献1】

特開2002-23958号

ビジネス向けストレージシステムには高信頼性が求められており、複数プロセッサの間でお互いの監視を行い、あるプロセッサに障害が発生したときには、他のプロセッサが障害プロセッサの作業を肩代わりすることで、通常動作を続けるシステムとなっている。複数プロセッサ間での監視は、一般的に現在時刻等の情報を定期的にやり取りするハートビートと呼ばれる通信を行う。一定時間ハートビート通信の応答がなくなると、対象プロセッサがダウンしたと考え、作業の移行を行う。また、ハートビート通信に限らず、システムの構成情報といった内部プロセッサ間で通信の行われるデータが多い。これらの通信は、一般的には外部ネットワークを用いてデータのやりとりが行われる。外部ネットワークを用いずに内部通信を行うには、内部ネットワークを用いる方法があるが、スペースの関係からあまり高速な内部ネットワークを張るのは難しい。

40

##### 【0003】

50

**【発明が解決しようとする課題】**

複数プロセッサ間の通信に対して、外部ネットワークを用いる場合、通信データが外に流れると、セキュリティ上問題になりやすい。外部ネットワークを使用しない場合には、特別に内部に専用ネットワークを結ぶ必要があり、不要なコストがかかる。本発明では、複数のプロセッサから共通してアクセスできるディスクキャッシュを介してデータのやり取りを行うことで、外部ネットワークを用いないセキュアな通信を、改めて専用ネットワークを張ることなく実現することを目的とする。ディスクキャッシュを介して複数プロセッサ間で通信を行う従来技術に特開2002-23958がある。これは、メインフレームホストが通信データをディスクキャッシュ上に常駐させ、オープン系ホストがそれを読み込むことで通信を行う方式である。この方式では、メインフレームホストとオープン系ホストは異なるシステムであり、オープン系ホストとディスクキャッシュとの間の接続を外部ネットワークを介する点で、セキュリティ上の問題が発生しうる。また、通信の方向もメインフレームホストからオープン系ホストへの一方で、双方向通信が不可能である。10

**【0004】****【課題を解決するための手段】**

上記課題を解決するために本願発明の提案する通信方法は、複数のCPUと、そのCPUからアクセス可能な共有メモリを含むコンピュータシステムにおける通信方法において、共有メモリの一部を通信バッファとして割当てるステップと、複数のCPUのうちいずれかが当該通信バッファに転送データを格納済みであるかどうかを判定するステップと、通信バッファに転送データが格納されていなければ複数のCPUのうちいずれかのCPUが通信バッファに転送データを格納するステップと、バッファにデータが格納されていれば複数のCPUのうちいずれかのCPUがバッファから転送データを読み込むステップと、複数のCPUのうちいずれかのCPUがバッファから転送データを読み込んだ後、バッファを転送データが格納されていない状態とするステップを有する。20

**【0005】**

例えばディスクキャッシュは共有メモリとして用いるのに好適である。本発明では、共有メモリの一部を通信バッファとして割り当て、通信バッファを例えばデータ転送領域と転送フラグ領域の組として管理する。データ転送領域にはCPU間で転送されるべきデータが格納される。転送フラグ領域はデータ転送領域に転送されるべきデータがあるかどうか示すフラグを格納する。例えば何もデータが入っていない領域のフラグは「0」とする。ある領域にCPUからデータが転送され、かつ、そのデータがまだ他のCPUによって読み出されていない場合、その領域のフラグは「1」とする。その領域のデータが他のCPUによって読み出された場合、その領域のフラグは「0」とする。すなわち、この例では、転送すべきデータが格納されている領域は「1」、転送すべきデータが格納されておらず、CPUが通信バッファとして使用できる領域を「0」としている。30

**【0006】**

CPU間でデータを転送しようとする場合には、まず、転送するデータを格納できる通信バッファの領域を検出する。上の例ではフラグが「0」の領域を探す。そして、フラグが「0」の領域に転送しようとするデータを格納し、フラグを「1」とする。このフラグをチェックすることにより、当該領域に他のデータが上書きされることを防止する。各領域のフラグは定期的にあるいは所定命令を契機としてチェックされ、フラグが「1」の領域のデータは転送先CPUへ読み出される。読み出しの後、フラグは「0」とされて通信バッファ領域はクリアされる。40

**【0007】**

これらの動作はCPUあるいはI/Oプロセッサ上で動作するソフトウェアにより実現可能である。さらに、専用のハードウェアで構成することも可能である。

**【0008】**

本発明の他の例である情報処理装置は、I/O命令を発行する少なくとも二つ以上のCPUと、CPUから直接的または間接的にアクセス可能なディスク装置と、ディスク装置のデータの少なくとも一部のコピーを格納するディスクキャッシュと、ディスク装置およびディスク

10

20

30

40

50

キャッシュを制御するためのI/Oプロセッサとを有し、CPUは前記I/Oプロセッサに対してディスクキャッシュの一部領域の割当要求を行う機能を有し、I/Oプロセッサは割当要求を受けるとディスクキャッシュの一部の領域を通信バッファとして割り当てる機能を有し、任意のCPUが通信バッファにデータを書き込み、書き込んだCPUとは異なるCPUがデータを読み込む通信手段を有する。

#### 【0009】

通信手段として、CPUまたはI/Oプロセッサは、通信バッファにこれから転送されるデータが格納されているかどうかを判定する手段と、通信バッファにこれから転送されるデータが格納されていなければ任意のCPUから転送データを通信バッファに転送して格納する手段と、通信バッファにこれから転送されるデータが格納されていれば任意のCPUが通信バッファから転送データを読み込む手段と、バッファをクリアする手段を有する。ディスクキャッシュは、プロセッサがディスクI/Oを行った際に最新データを高速なメモリ上に配置し、アクセス性能を向上するためのものである。そのため、プロセッサからディスクキャッシュにアクセスするには、ディスクI/Oを行う必要があり、またその際に書き込まれるキャッシュのアドレスを指定することはできない。本発明の一例では、特別なコマンドをディスク制御用プロセッサに対して発行することで、ディスクにアクセスすることなしにディスクキャッシュの特定領域にアクセスすることを可能とし、その特定領域に送信側プロセッサが書き込んだデータを受信側プロセッサが読み込むことで、複数プロセッサ間での通信を行う。それにより高速でセキュアな通信を可能とする。

10

#### 【0010】

##### 【発明の実施の形態】

20

##### [実施例1]

本発明の第一の実施例を説明する。

図1は本発明を説明する概念図である。101～103はCPU、111～113はディスクおよびディスクキャッシュを制御するI/O用プロセッサ(IOP)、121～123はI/O用BUS、131はディスクキャッシュ、141～142はディスクである。全てのCPU101～103は共通のディスクキャッシュ131とBUS121～123を介して接続されている。これら全体で一つのストレージ装置を構成している。CPU101～103がI/Oを行うと、IOP111～113はディスクキャッシュ131に対するキャッシュ制御を行い、ディスク141～142にアクセスする。このとき、CPU101～103はディスクキャッシュ131を制御することはできず、その特定領域を指定してアクセスすることはできない。そこでまず、CPU101～103はIOP111～113に対してディスクキャッシュ131の一部領域の割当要求を行う(151)。IOP111～113は割当要求を受けるとディスクキャッシュ131の一部の領域132をCPU101～103のメモリ空間に割り当てるにより、CPU101～103が直接ディスクキャッシュ131にアクセスすることを可能とする(152)。CPU101～103が直接ディスクキャッシュ131にアクセスすることが可能になると、各CPU間、例えばCPU101とCPU102との間の通信は、CPU101がディスクキャッシュ131の特定領域に書き込んだデータをCPU102が読み込むことで実現可能となる(153)。このとき、ディスクキャッシュ131に書き込んだデータはディスク141～142に反映する必要はない。

30

#### 【0011】

図2はCPUのアドレス空間の概要を示す。CPU101～103のアドレス空間201はCPUのメモリ領域202を指している。それに対して、CPU101～103がIOP111～113に割当要求を行うと、ディスクキャッシュ131の一部領域132をCPUのアドレス空間201の一部領域203に割り当てる。これにより、CPU101～103はアドレス空間201のうちの特定領域203にアクセスすることで、直接ディスクキャッシュ内の領域132にデータの読み書きが可能となる。IOP111～113を介さずに、CPU101～103がディスクキャッシュ131に直接アクセスするため、ディスクキャッシュ131へのアクセスはディスクI/Oコマンドで行うのではなく、メモリへのアクセスコマンドで行う。そのためディスクキャッシュ131にデータを書き込んでもディスク141～142にデータは反映されない。また、CPU101～103が直接アクセスできるディスクキャッシュ内の領域132は、ディスクI/O時にキャッシュ領域として使用できない。

40

上記の例は、ディスクキャッシュの一部をCPUのアドレス空間にマッピングすることで、C

50

PUから直接ディスクキャッシュにデータの読み書きを行う方式である。

#### 【0012】

図3は他の方式の例である。図3のように通信に使用するディスクキャッシュのアドレスを指定して、DMAを用いてデータの転送を行う方式も可能である。CPU101はIOP111に対して、ディスクキャッシュの一部の領域を通信領域として割り当てる特別のI/Oコマンドで依頼する(301)。通常の装置ではI/OプロセッサへのI/Oコマンドはディスクアクセスのためのものであるが、本発明では、ディスクアクセスではなくCPU間データ転送用のI/Oコマンドを用いる。それを受けたて、IOP111はディスクキャッシュの一部領域132を通信領域として割り当て、CPU101に対してそのアドレスとサイズを通知する(302)。以降、通信領域として割り当てられた領域は、CPU101が自由に使用でき、アドレスとサイズを指定して通信データをDMA転送することができる(303)。通信領域として割り当てる領域は、ディスクI/Oに対するキャッシュ領域として使用しない特別な領域であるか、あるいは通常の領域であれば以降、キャッシュ領域として使用しないようIOP111で制御することが必要である。図4はCPU間通信の概要を示す。CPUから直接アクセスするディスクキャッシュ領域132に通信バッファ411～416を用意する。本実施例では、全CPU間で双方の通信を行うため、それぞれの通信用に別の通信バッファを設けた。例えば、CPU101からCPU102にデータ転送を行う場合、通信バッファ411を用いる。CPU101は転送データ401を通信バッファ311に書込み、CPU102は通信バッファ411のデータを受信領域402に読み込む。10

#### 【0013】

図5は通信バッファの概念を示す。通信バッファは図5のように内部が複数のメッセージキューに分かれしており、先頭から順にラウンドロビンに使用する。各メッセージキューには転送フラグ領域501と転送データを書き込む領域502が存在し、送信CPU101は転送フラグ領域501にフラグがたっていないを確認して転送データ領域502にデータを転送し、転送完了後転送フラグ領域501にフラグをたてる。受信CPU102は転送フラグ領域501にフラグがたっているメッセージキューを検索し、転送データ領域502から受信バッファ402にデータを読み込む。読み込み完了後転送フラグ領域501のフラグを落とす。メッセージキューは先頭から順に使用するため、送受信CPUとも前回使用したキューの位置を記憶しておくことにより、通信時に確認すべき転送フラグ領域501の位置は一意に定まる。20

#### 【0014】

図6は本実施例の送信処理のフローチャートである。送信CPUが転送フラグ領域を読み込み(601)、メッセージキューの使用状況を確認する(602)。メッセージキューが全て使用中であれば、タイムアウトするまでキューが開放されるのをチェックし(606)、タイムアウト時間内にキューが開放されなければネットワークBUSYでエラー終了となる(607)。一方、602において、空きバッファが見つかれば転送データをバッファに書き込む(603)。転送データ書き込み後、転送フラグ領域にフラグをセットし(604)、送信を完了する(605)。30

#### 【0015】

図7に受信処理のフローチャートを示す。受信CPUは転送フラグ領域を読み込み(701)、転送データが書き込まれたか確認する(702)。転送フラグがたっていないなければそのまま終了し(705)、たっていれば当該バッファより転送データを読み込む(703)。読み込み完了後転送フラグをクリアし(704)、受信処理を終了する(706)。受信CPUが転送フラグ領域の読み込みを行う契機は、定期的に読み込みを繰り返す方式と、ディスクキャッシュ上の通信領域にデータが書き込まれるとCPUにハードまたはソフト割り込みをあげる方式がある。40

#### [実施例2]

次に本発明の第二の実施例について説明する。第二の実施例は、CPUが直接ディスクキャッシュにアクセスせず、ディスクキャッシュの制御をIOPに依頼する点が特徴的である。すなわちCPUは転送データのアドレスをIOPに通知し、IOPがディスクキャッシュを介して受信CPUにデータを転送する方式である。

ディスクキャッシュ上の通信バッファの仕様は図4、5と同一である。CPUからは直接ディスクキャッシュをアクセスできないため、メッセージキューへの書き込みはIOPを経由して行う。50

## 【0016】

図8に本実施例の概要を示す。CPU101は転送データの格納されるメモリ801のアドレスをIOP111に通知する(821)。転送データはCPU101に内蔵されるあるいは外付けされるメモリ801内に格納されている。CPU101からの通知を受けたIOP111はメモリ801から転送データをディスクキャッシュ上の通信バッファ811に転送する(822)。受信CPU102はIOP112に転送データを受信するメモリ802のアドレスを通知する(823)。CPU102からの通知を受けたIOP112は通信バッファ811からメモリ802にデータ転送する(824)。送受信IOP間のメッセージキューの管理方法は第一の実施例におけるCPU間のメッセージキューの管理方法と同一である。すなわち、送信IOP111は通信バッファ811内の転送フラグを確認し、空きバッファを検索する。検索した空きバッファにデータの書き込みを行い、転送フラグにフラグを立てる。受信IOP112は転送フラグを確認し、データ到着を検出すると通信バッファからデータを読み込み受信バッファ802に格納する。最後に転送フラグを落とす。CPUのローカルメモリ801-803にディスクキャッシュ上のメッセージキューと同様に、送受信先ごとにメッセージキューを用意し、先頭から順にキューを使用することとすれば、CPUからIOPに対して発行する通信要求には送受信データのアドレスを指定する必要はない。

例えば、CPU101のローカルメモリ801には、送信キューとしてCPU1からCPU2への送信キュー（ディスクキャッシュ132の811に対応する）、CPU1からCPU3への送信キュー（ディスクキャッシュ132の816に対応する）、CPU1からCPU4への送信キュー・・・を設け、また、受信キューとしてCPU2からCPU1への受信キュー（ディスクキャッシュ132の812に対応する）、CPU3からCPU1への受信キュー（ディスクキャッシュ132の815に対応する）、CPU4からCPU1への受信キュー・・・を設ける。他のローカルメモリ802、803、...も同様である。

この場合、送信CPUが送信要求をIOPに通知すると、IOPはCPUローカルメモリ801のキューからディスクキャッシュ132の対応するキューにデータ転送する。このとき、ローカルメモリ801とディスクキャッシュ132の各キューはラウンドロビンで順番に使用される。

## 【0017】

また、受信IOPはディスクキャッシュ132上のメッセージキューが更新されたのを契機として受信バッファに転送データを格納する。このとき、ディスクキャッシュ132のキューからCPUローカルメモリ802の対応するキューにデータ転送する。このときも、各キューはラウンドロビンで順番に使用される。その後、受信CPUにデータ格納完了を通知する。

以上の構成によれば、キューは順番に使用されるので、CPUからIOPに対して発行する通信要求には送受信データのアドレスを指定する必要はない。

CPUからIOPへの通信要求は特別なI/Oコマンドを用いて行われる。通常I/Oコマンドは、コマンド種別(READ/WRITE)、CPUのローカルメモリアドレス、ディスクのアドレス等の情報を含んでいる。これらの情報を元にIOPはCPUローカルメモリアドレスにあるデータをディスクアドレスに書き込み、あるいは逆に読み込みを行う。本実施例では、コマンドのI/Fは通常I/Oコマンドと同一のまま、コマンド種別に通常のREAD/WRITEではなく、送信/受信を示す特別なコマンド種別を指定することで、IOPは通信要求であると判定することが可能である。本発明では通信用のディスクキャッシュに書き込まれるデータは、通信相手CPUに対して転送するためのもので、ディスクに反映する必要はない。したがって、IOPは通信要求を示す特別なコマンドが渡されると、ディスクにI/Oせずにディスクキャッシュにのみデータを反映させるという特別な処理を行う。受信CPUから受信要求コマンドが渡されると、同様にディスクからI/Oせずにキャッシュから直接データを読み込む。本実施例において、CPUからIOPに発行する通信要求は特別なI/Oコマンドとするのではなく、特別なアドレスへのI/Oコマンドを発行することで通信要求とすること、あるいは通信デバイスとして通常のネットワークデバイスではなく、特別に用意した仮想デバイスを指定して通信を発行することで通信要求とすることもできる。

## 【0018】

上記二つの実施例では、送受信処理をそれぞれ独立に扱い、受信処理の発生契機を受信関数のコールまたはデータ到着の認識としている。それに対して、第三の実施例として送信CPUが送信開始後、受信CPUに対して通信開始を通知する方式もある。ディスクキャッシュ

10

20

30

40

50

の特定領域にデータを書き込むと受信CPUに割込みが発生するする仕組みを用意し、送信CPUが転送データをディスクキャッシュに書き込み後、割り込み発生領域に書き込むことで受信CPUに通知する。あるいは、ディスクキャッシュより小容量で高速なネットワークあるいは共有メモリを用意し、送信開始通知を高速バスで送ることで受信CPUに通知する。割込みまたは別バスで送信開始通知を受けた受信CPUは、それを契機に受信処理を開始する。

#### 【0019】

##### 【発明の効果】

本発明では、ディスクキャッシュの一部の領域をCPUから直接アクセスすることを可能とし、ディスクキャッシュ上に通信用のバッファを配置する。送信CPUがディスクキャッシュ上の通信バッファに書き込んだデータを受信CPUが読み込むことで通信を実現できる。  
ディスクキャッシュは、ディスクI/O高速化のため標準的に用意されており、全CPUから共通してアクセスすることが可能である。このため新たな装置を用意することなく、全CPU間での双方向通信が可能となる。また、内部バスを介してデータ転送を行うため、外部からデータが見えず、セキュアな通信が可能となる。通信コマンドは一般的なI/Oコマンドと同一フォーマットであり、新たない/Fを必要としない。

10

##### 【図面の簡単な説明】

【図1】本発明の実施例を示した概略ブロック図である。

【図2】ディスクキャッシュのCPUアドレス空間へのマッピングを示す概念図である。

【図3】ディスクキャッシュへのアクセスをDMAで行う方式を示すブロック図である。

20

【図4】実施例のデータ転送方法を示すブロック図である。

【図5】ディスクキャッシュ上のメッセージキューの構成図である。

【図6】実施例の送信処理の流れ図である。

【図7】実施例の受信処理の流れ図である。

【図8】別の実施例におけるデータ転送方法を示すブロック図である。

##### 【符号の説明】

101～103…CPU

111～113…IOP

121～123…データバス

131…ディスクキャッシュ

30

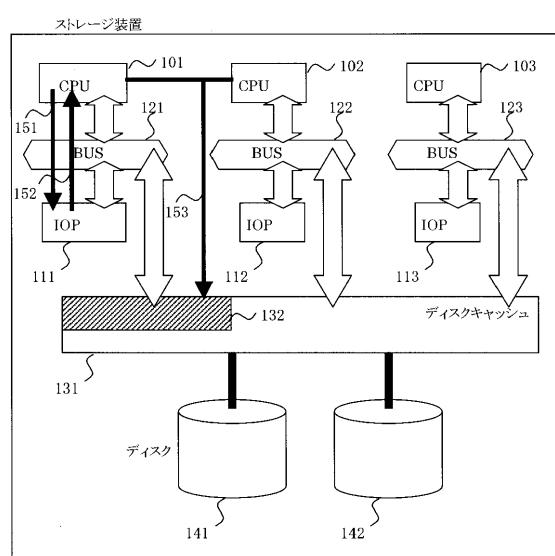
132…ディスクキャッシュのうち通信に使用する領域

141～142…ディスク

151～153…通信処理の流れ。

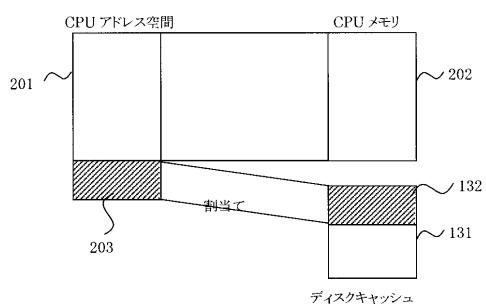
【図1】

図1



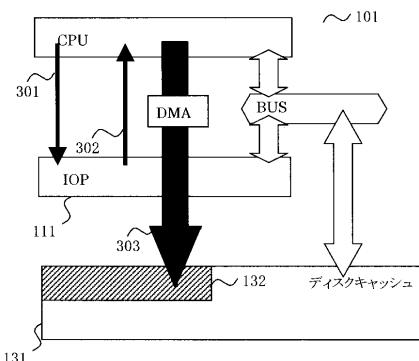
【図2】

図2



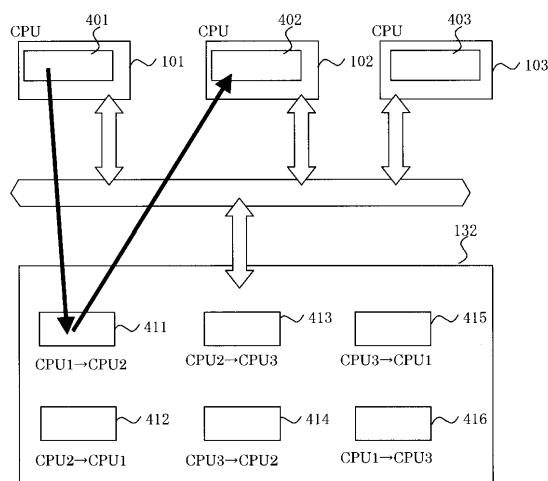
【図3】

図3



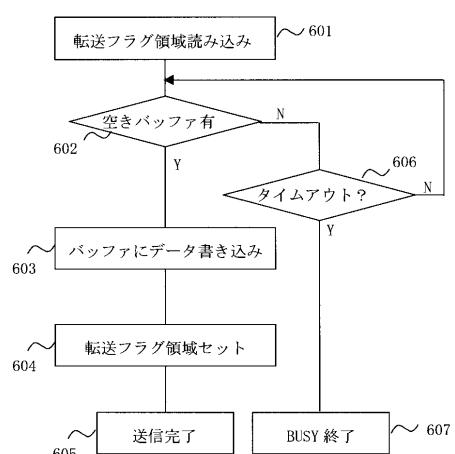
【図4】

図4



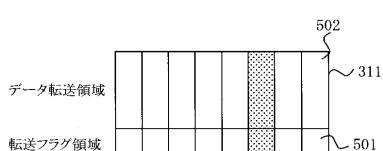
【図6】

図6



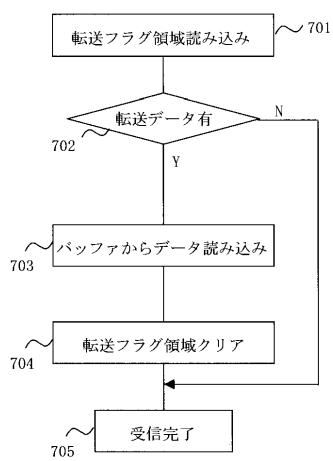
【図5】

図5



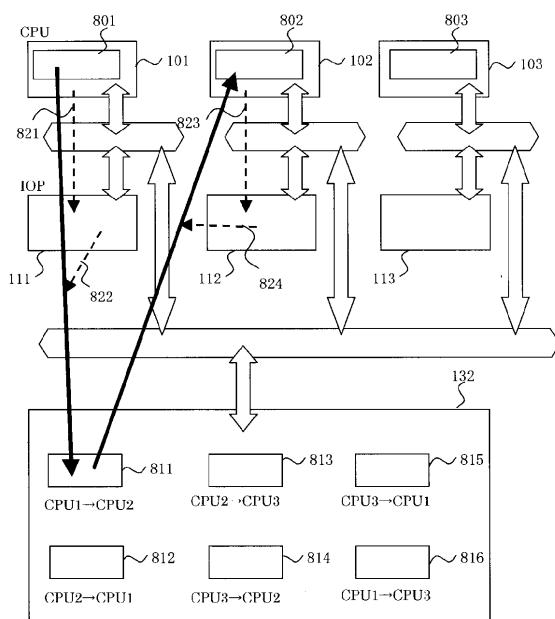
【図7】

図7



【図8】

図8



---

フロントページの続き

(51)Int.Cl.

F I

G 0 6 F	12/08	5 5 5
G 0 6 F	12/08	5 5 7
G 0 6 F	12/10	5 9 9

(56)参考文献 特開平11-120156(JP,A)

特開平04-067240(JP,A)

特開平02-016651(JP,A)

特開2002-023958(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 15/16-15/177

G06F 3/06-3/08

G06F 12/08-12/12