

(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2010-86227

(P2010-86227A)

(43) 公開日 平成22年4月15日 (2010.4.15)

(51) Int.Cl.		F I			テーマコード (参考)
G06F 11/20	(2006.01)	G06F 11/20	310C		5B005
G06F 12/10	(2006.01)	G06F 12/10	559		5B034
		G06F 12/10	541		

審査請求 未請求 請求項の数 10 O L (全 33 頁)

(21) 出願番号	特願2008-253793 (P2008-253793)	(71) 出願人	000004226
(22) 出願日	平成20年9月30日 (2008.9.30)		日本電信電話株式会社
			東京都千代田区大手町二丁目3番1号
(特許庁注：以下のものは登録商標)		(74) 代理人	100083552
1. イーサネット			弁理士 秋田 収喜
		(74) 代理人	100103746
			弁理士 近野 恵一
		(74) 代理人	100119703
			弁理士 井上 雅夫
		(72) 発明者	小倉 毅
			東京都千代田区大手町二丁目3番1号 日
			本電信電話株式会社内
		Fターム (参考)	5B005 JJ01 KK02 KK16 MM35 RR01
			5B034 BB02 CC05

(54) 【発明の名称】 計算機間相互結合網における通信経路の冗長化と切り替え方法、この方法を実現するサーバ装置、そのサーバモジュール、および、そのプログラム

(57) 【要約】

【課題】 通信経路上で障害が発生した場合の迅速な通信経路の切り替え動作を維持しながら、従来の自明な通信経路冗長化方式が持つ問題を解決する。

【解決手段】 正常動作時には、冗長化された通信経路のうちの1つだけを使用してホストA、B間でのデータ転送を行う。ホストAでは、データがディスクから読み出され主メモリ上のセグメント (RDMA領域) に格納されていくとともに、LMR群1を用いて通信経路1経由でRDMAによりそれらのデータがホストBへ転送される。ホストBでは、LMR群1を用いて通信経路1から受信したデータを主メモリに格納する。通信経路1に障害が発生すると、ホストA、BはRDMAに使用するLMR群を1から2へ切り替え、まだホストBへ転送していない主メモリ上のセグメントから、RDMAによるデータ転送を再開する。

【選択図】 図10

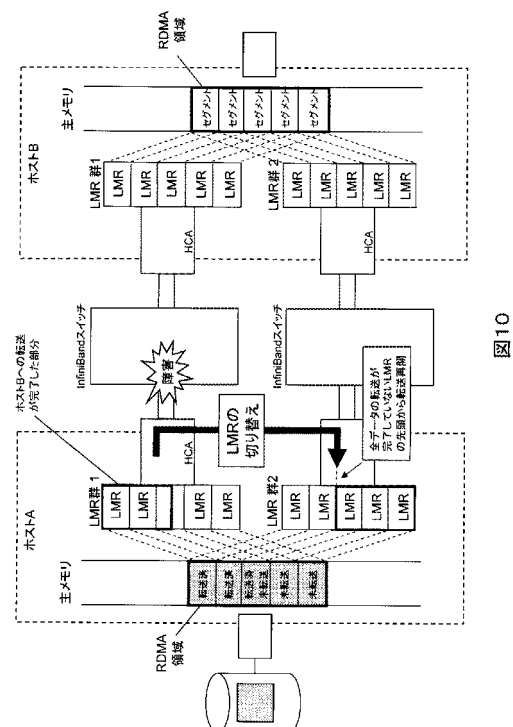


図10

【特許請求の範囲】

【請求項 1】

計算機間のコネクション設定機構および当該計算機の主メモリ領域の抽象化機構を持つ通信方式を実行可能な相互結合網を用いて計算機を複数台接続し、さらに、その相互結合網が D (≥ 2) 個の系に冗長化され各計算機がそれぞれの系に接続するための D 本の通信リンクを有する構成のサーバ装置における通信経路の冗長化と切り替え方法であって、

前記各計算機は、

自身の通信リンク i ($1 \leq i \leq D$) と対向の計算機の通信リンク i との間にコネクションを設定して合計 D 本の冗長な通信経路を確保する経路冗長化手段と、

通信対象となるデータの格納領域として主メモリ内に確保した 1 つのデータ格納手段と

、
前記各通信経路に対応し全て前記データ格納手段に対応づけられた D 個のメモリ領域抽象化手段を有し、

ある 1 つの前記メモリ領域抽象化手段を経由することによりそれに対応した通信経路を使用して対向の計算機との間で前記データ格納手段に含まれるデータの転送を行っている最中に当該通信経路に障害が発生した場合に、使用するメモリ領域抽象化手段を他の通信経路に対応したメモリ領域抽象化手段に切り替えるだけで当該データ転送に使用する通信経路を切り換えて当該データ転送を継続する、通信経路の冗長化と切り替え方法。

【請求項 2】

請求項 1 に記載の通信経路の冗長化と切り替え方法において、

前記サーバ装置に対し、 D (≥ 2) 個の系に冗長化された相互結合網の任意の系の間で通信が可能となるように相互結合網間通信リンクを追加したサーバ装置において、

前記各計算機に対し、自身の計算機の通信リンク i ($1 \leq i \leq D$) と対向の計算機の全ての通信リンクとの間にコネクションを設定して合計 D^2 本の冗長な通信経路を確保するように前記経路冗長化手段を変更し、また、前記メモリ領域抽象化手段を前記各通信経路に対応した D^2 個持たせるように変更した、通信経路の冗長化と切り替え方法。

【請求項 3】

請求項 1 または 2 に記載の通信経路の冗長化と切り替え方法において、

前記各計算機は、各通信経路に対応し、当該通信経路を経由して対向の計算機と軽量なメッセージ交換を行い、データ転送に通常要する時間よりもわずかに長い時間をデータ転送の規定時間とし、さらにこれよりも短い時間を前記メッセージ交換の規定時間として、前記メッセージ交換がこの規定時間以内に終了できた場合にはその通信経路の状態として正常状態を記録した後同じ処理を繰り返し、規定時間以内に終了できなかった場合にはその通信経路の状態として異常状態を記録して処理を終了する機能を有する通信経路障害検出手段を有し、

データ転送がその規定時間以内に終了できなかった場合には、前記通信経路の状態を参照して、正常状態であればデータ転送の終了を次の規定時間が経過するまで継続して待ち、異常状態であれば正常状態である別の通信経路に切り替えて当該データ転送を継続する、通信経路の冗長化と切り替え方法。

【請求項 4】

請求項 3 に記載の通信経路の冗長化と切り替え方法において、

前記各計算機は、前記通信経路障害検出手段に対し、通信経路の状態として異常状態を記録した後も前記メッセージ交換を継続して行い、再びメッセージ交換をその規定時間以内に終了できるようになった場合には通信経路の状態を正常状態に戻すように機能を変更した通信経路障害・回復検出手段を有し、

これにより一度障害が発生した後に通信が回復した通信経路を再使用可能にする、通信経路の冗長化と切り替え方法。

【請求項 5】

計算機間のコネクション設定機構および当該計算機の主メモリ領域の抽象化機構を持つ通信方式を実行可能な相互結合網を用いて計算機を複数台接続し、さらに、その相互結合

10

20

30

40

50

網が D (≥ 2) 個の系に冗長化され各計算機がそれぞれの系に接続するための D 本の通信リンクを有する構成のサーバ装置であって、

前記各計算機は、

自身の通信リンク i ($1 \leq i \leq D$) と対向の計算機の通信リンク i との間にコネクションを設定して合計 D 本の冗長な通信経路を確保する経路冗長化手段と、

通信対象となるデータの格納領域として主メモリ内に確保した 1 つのデータ格納手段と、

前記各通信経路に対応し全て前記データ格納手段に対応づけられた D 個のメモリ領域抽象化手段を有し、

ある 1 つの前記メモリ領域抽象化手段を経由することによりそれに対応した通信経路を使用して対向の計算機との間で前記データ格納手段に含まれるデータの転送を行っている最中に当該通信経路に障害が発生した場合に、使用するメモリ領域抽象化手段を他の通信経路に対応したメモリ領域抽象化手段に切り替えるだけで当該データ転送に使用する通信経路を切り換えて当該データ転送を継続する、サーバ装置。

【請求項 6】

請求項 5 に記載のサーバ装置において、

前記サーバ装置に対し、 D (≥ 2) 個の系に冗長化された相互結合網の任意の系の間で通信が可能となるように相互結合網間通信リンクを追加したサーバ装置において、

前記各計算機に対し、自身の計算機の通信リンク i ($1 \leq i \leq D$) と対向の計算機の全ての通信リンクとの間にコネクションを設定して合計 D^2 本の冗長な通信経路を確保するように前記経路冗長化手段を変更し、また、前記メモリ領域抽象化手段を前記各通信経路に対応した D^2 個持たせるように変更した、サーバ装置。

【請求項 7】

計算機間のコネクション設定機構および当該計算機の主メモリ領域の抽象化機構を持つ通信方式を実行可能な相互結合網を用いて複数台の計算機を接続し、さらに、その相互結合網が D (≥ 2) 個の系に冗長化され各計算機がそれぞれの系に接続するための D 本の通信リンクを有し、前記複数台の計算機は外部装置と直接通信する通信サーバモジュールと蓄積装置を制御する蓄積サーバモジュールから構成されるサーバ装置における通信サーバモジュールであって、

自身の通信リンク i ($1 \leq i \leq D$) と前記蓄積サーバモジュールの通信リンク i との間にコネクションを設定して合計 D 本の冗長な通信経路を確保する経路冗長化手段と、

通信対象となるデータの格納領域として主メモリ内に確保した 1 つのデータ格納手段と、

前記各通信経路に対応し全て前記データ格納手段に対応づけられた D 個のメモリ領域抽象化手段を有し、

ある 1 つの前記メモリ領域抽象化手段を経由することによりそれに対応した通信経路を使用して対向の計算機との間で前記データ格納手段に含まれるデータの転送を行っている最中に当該通信経路に障害が発生した場合に、使用するメモリ領域抽象化手段を他の通信経路に対応したメモリ領域抽象化手段に切り替えるだけで当該データ転送に使用する通信経路を切り換えて当該データ転送を継続する、通信サーバモジュール。

【請求項 8】

計算機間のコネクション設定機構および当該計算機の主メモリ領域の抽象化機構を持つ通信方式を実行可能な相互結合網を用いて複数台の計算機を接続し、さらに、その相互結合網が D (≥ 2) 個の系に冗長化され各計算機がそれぞれの系に接続するための D 本の通信リンクを有し、前記複数台の計算機は外部装置と直接通信する通信サーバモジュールと蓄積装置を制御する蓄積サーバモジュールから構成されるサーバ装置における蓄積サーバモジュールであって、

自身の通信リンク i ($1 \leq i \leq D$) と前記通信サーバモジュールの通信リンク i との間にコネクションを設定して合計 D 本の冗長な通信経路を確保する経路冗長化手段と、

通信対象となるデータの格納領域として主メモリ内に確保した 1 つのデータ格納手段と

10

20

30

40

50

、
前記各通信経路に対応し全て前記データ格納手段に対応づけられたD個のメモリ領域抽象化手段を有し、

ある1つの前記メモリ領域抽象化手段を経由することによりそれに対応した通信経路を使用して対向の計算機との間で前記データ格納手段に含まれるデータの転送を行っている最中に当該通信経路に障害が発生した場合に、使用するメモリ領域抽象化手段を他の通信経路に対応したメモリ領域抽象化手段に切り替えるだけで当該データ転送に使用する通信経路を切り換えて当該データ転送を継続する、蓄積サーバモジュール。

【請求項9】

請求項7に記載の通信サーバモジュールとしてコンピュータを機能させるためのプログラム。 10

【請求項10】

請求項8に記載の蓄積サーバモジュールとしてコンピュータを機能させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

PC (Personal Computer) などの計算機を複数台接続して1つの高速な計算機システムを構築するPCクラスタリング分野において、計算機間を接続する相互結合網の構成を冗長化して通信経路の耐障害性を向上させる、PCクラスタシステム内部通信の高信頼化手法に関する。 20

【背景技術】

【0002】

処理能力が低いかわりに安価であるPC等の小型計算機を多数接続して、1台の大型で高速な計算機システムと同等の処理能力を有するシステムを経済的に構築する、PCクラスタリングとよばれる技術がある。このPCクラスタリング技術は、科学技術計算システム、データセンター等における各種サーバ、大容量ストレージシステムなどの構築手法として幅広い分野で用いられている。

【0003】

PCクラスタリングにおいて、PC間の接続に用いられる物理伝送媒体を相互結合網と呼ぶ。相互結合網には様々な種類があり、それぞれが持つ通信機能の特色を生かした使い分けがなされている。相互結合網の具体的な例としては、Myrinet, Fibre Channel, InfiniBandなどがある。さらに通常は、1種類の相互結合網においても、その上位レイヤでは複数の種類の通信方式(プロトコル)を動作させることができる。図1にPCクラスタシステムの概要を示す。 30

【0004】

これら数ある相互結合網の中でも、高速性、低価格性、使い易さ等が評価され、近年のPCクラスタシステムにおいて最も広く用いられているのがInfiniBand(非特許文献1)である。そして、このInfiniBandの上位レイヤではTCP/IP, SDP, MPI等の様々な種類のプロトコルが動作可能であるが、その中の代表的なプロトコルの1つにuDAPL(user direct access programming library)(非特許文献2)がある。 40

【0005】

uDAPLとは、InfiniBand等の幾つかの相互結合網自体が持つRDMA(remote direct memory access)機能を、相互結合網の種類に依らない共通のソフトウェアインタフェースを介して利用できるようにするための統一されたAPI(application interface)、および、データ転送プロトコルを提供するソフトウェア体系である。ここで、RDMAとは、従来から広く用いられている、1台の計算機における入出力(I/O)処理の高速化技術であるDMA(direct memory access)を、LAN(local area network)や相互結合網で接続された遠隔の計算機間の通信に拡張したものである。 50

【0006】

DMAでは、計算機に直結された外部装置との間のデータの入出力（I/O）の際に、CPUをほとんど使用せずに当該外部装置のコントローラが自計算機の主メモリに直接アクセスしてデータ転送を行うことで、計算機にかかる処理負荷の低減やデータ入出力の高速化を図る。

【0007】

RDMAにおいても同様に、遠隔の計算機どうしの主メモリ間でCPUをほとんど使用しないデータ転送が行えるため、非常に低負荷かつ高速な計算機間通信が可能となる。実際、uDAPLはInfiniBandの上位レイヤで動作するプロトコル群の中でも最も高速な物の1つであり、重要性が高まっている。

10

【0008】

RDMAでは、データ送信や受信の対象となる計算機の主メモリの領域をあらかじめ通信相手に対して公開し、データ転送経路を確立した後に通信を開始するのが本質的な動作形態となる。この点において、その他の一般的なデータ転送方式、すなわち、送信側では受信側の最終的なデータの格納場所を認識せずにデータを送信し、受信側ではデータを受信するたびにアドホックに格納場所を決定するような方式とは異なっている。

【0009】

後者のような方式では、計算機間であらかじめ通信経路を確立しない非コネクション型通信が可能であるのに対し、前者のRDMAにおいては、計算機間で事前に通信経路を確立するコネクション型通信が前提となる。

20

【0010】

さらに、データの送信／受信処理の対象となるメモリ領域を明示的に指定する必要がある。

【0011】

したがってuDAPLも、このようなRDMAの本質的な動作形態を反映したもの、すなわち、コネクション設定機構やデータ転送の対象となる主メモリ領域の抽象化機構等を中心としたAPIとなっている。以下にuDAPLのAPIの概要を説明する。

【0012】

まず、uDAPLのAPIで用いられる主なオブジェクト群を図2に示す。InfiniBand HCA (Host Channel Adapter) を搭載したホストA（以降、PC等の小型計算機をホストと呼ぶ）とホストBとがInfiniBandスイッチ経由で接続されており、上記ホストの主メモリ間でRDMAによるデータ転送を行うことを前提とした図である。なお、ここではホストAがRDMAのイニシエータ、すなわち、RDMAによるデータ転送を発動する側となる場合を示している。

30

【0013】

前述のように、uDAPLはコネクション型の通信方式を採用している。図2のEP (End Point) は、通信主体となるアプリケーションプロセス間のコネクションの端点を抽象化したオブジェクトである。コネクションはEP間で確立され、1対1接続のみが可能である。アプリケーションプロセスは通信処理を起動する際、自身が保持するEPのうちの特定の1つを指定する。これにより、その通信に使われるコネクション（と通信の相手となるEP）が特定される。

40

【0014】

なお、本明細書では、ホストのHCAとInfiniBandスイッチ間の物理的な1本の接続回線を通信リンクと呼ぶ。また、EP間で確立された論理的な通信経路をコネクションと呼ぶ。

【0015】

LMR (Local Memory Region) は、通信動作の対象となる主メモリの領域を抽象的に表現するためのオブジェクトである。1つのLMRは、単一の連続した仮想的なメモリ領域を表す。アプリケーションプロセスはこのLMRが表す仮想的な連続メモリ領域の全部、あるいは一部を指定して通信処理を起動する。LMRはその使用に先立って、それが対

50

応づけられている（マッピングされている）主メモリの実際の物理的な領域が決定されている。そのため、LMRの指定は当該通信処理の対象となる主メモリの領域を指定することを意味する。

【0016】

RMR（Remote Memory Region）は、自ホスト内のLMRにマッピングされた主メモリの領域に対する遠隔ホストからのアクセス制限に関する設定のためのオブジェクトである。RMRはLMRの全部、あるいは一部に対してバインドすることができ（図は1つのRMRをLMRの全領域にバインドした場合を示している）、バインドされたLMRの領域にマッピングされている主メモリの領域に対してRMRに付随して設定したアクセス制限が適用される。すなわち、LMRから任意の領域を切り出してアクセス制限を柔軟に設定するためのメモリウインドウを提供する。

10

【0017】

PZ（Protection Zone）は、各種オブジェクトをグループ化し、グループ外のオブジェクトからの操作を排除する手段を提供するオブジェクトである。上記EP、LMR、RMRはそれぞれ、その生成時にただ1つのPZと関連づけられる。あるEPを経由して実行される通信操作は、そのEPが属するPZと同じPZに属するLMRやRMRにマッピングされた主メモリの領域にしかアクセスできない。

【0018】

EVD（Event Dispatcher）は、uDAPLの各種操作の完了をイベントとして通知するためのオブジェクトである。前述のEPはその生成時において、メッセージ受信完了通知用EVD（receive EVD）、メッセージ送信/RDMA read/RDMA write/RMRのLMRへのマッピング完了通知用EVD（request EVD）、EP間コネクション確立通知用EVD（connect EVD）の3つのEVDと関連づけられる。アプリケーションプロセスは当該EPを介して行ったこれらの動作の完了を各EVDからのイベント通知により認識する。

20

【0019】

なお、上記EP、LMR、RMR、PZ、EVDの各オブジェクトは、その生成時に1つのInfiniBand HCAに関連づけられる。

【0020】

以上がuDAPLのAPIで用いられる主なオブジェクト群の説明である。次に、これらを用いて実際に通信を行う際の手順を説明する。例えば、ホストAがホストBの主メモリ内のデータをRDMAによって読み出し、自身の主メモリ内に格納する場合の手順は以下の（1）～（9）のようになる。本手順の概要を図3に示す。

30

【0021】

（1）ホストAは、RDMAによってアクセスすべきホストBのメモリ領域（ターゲットメモリ領域）に関する情報を要求する。この要求メッセージ自体はホストAの主メモリ内の要求送信バッファ内に格納されており、このバッファとマッピングされているLMR、および、EPを指定してメッセージ送信動作を起動する。（この動作はRDMAの機能を使用しない通常のメッセージ送信である。ホストAは、送信したメッセージがホストBの主メモリ上のどの場所に格納されるかを知らないことに留意）。

40

【0022】

（2）ホストAは、送信用EVDからの送信動作完了イベント通知を通して上記メッセージ送信が正常に完了した事を確認する。

【0023】

（3）ホストAは、メモリ情報受信バッファにマッピングされているLMRとEPを指定して、ホストBからのメモリ情報受信動作を起動する。

【0024】

（4）ホストAは、受信用EVDからの通知を通してホストBからのメモリ情報の受信完了を待つ。

【0025】

50

(5) ホストBは、自身の主メモリ内の要求受信バッファにマッピングされているLMR、および、EPを指定して、ホストAからのメモリ要求受信動作を起動する。

【0026】

(6) ホストBは、受信用EVDからの通知を通してホストAからのメモリ情報要求の受信完了を待つ。

【0027】

(7) ホストBは、ホストAからのメモリ情報要求を受信すると、要求受信バッファに格納されたメモリ要求の内容を確認し、ホストAからのRDMAによるメモリアクセスを許可する主メモリ領域(RDMA領域)に関する情報、具体的には、開始アドレス、レンジ、アクセスキーの3つ組(rmr__triplet)の送信動作を起動する。この時送信する内容は主メモリのメモリ情報送信バッファに格納しておき、この領域にマッピングされたLMRとEPを指定してメッセージ送信を行う。(この動作もRDMAとは異なる通常のメッセージ送信である)。

【0028】

(8) ホストBは、送信用EVDからの送信動作完了イベント通知を通して上記メッセージの送信完了を待つ。この完了を確認すると、(3)の動作に戻る。(ホストBはRDMA処理そのものの開始・終了については感知しない)。

【0029】

(9) ホストAは、受信用EVDからのイベント通知を通してホストBからのメモリ情報の受信を確認すると、メモリ情報受信バッファに格納された、ホストBからのメモリ情報(rmr__triplet)を用いて、(7)のRDMA動作の起動処理に移る。

【0030】

(10) 主メモリ上のRDMA領域にマッピングされたLMRを用いてその主メモリ領域をデータ受信用領域に指定し、上記rmr__tripletによりターゲットとなるホストBの主メモリ領域を指定して、また、EPを指定して、RDMA read処理を起動する。この処理を起動した後は、(9)で用いたのと同じの受信用EVDを用いてRDMA readの終了通知を待つ。

【0031】

(11) InfiniBandの物理層、および、リンクレイヤの機能によりRDMAによる実際のデータ転送が行われる。ホストBの主メモリ上のRDMA領域内のデータがInfiniBandで扱われる最大データ長(4Kbyte)より大きい場合、当該データは複数のパケットに分割されてホストAに転送される。ホストAでは、受信したパケットを主メモリ上のRDMA領域に格納していく。ホストA、Bにおいてこれらの処理がほとんどCPUの介在なく行われる。

【0032】

(12) ホストAは、EVDからの通知によりRDMA readの終了を確認すると、ホストBの当該LMRに相当するRDMA領域の全てのデータがホストAの当該LMRに相当するRDMA領域に転送されたと判断する。引き続き他のRDMA領域を使用してRDMA転送を続ける場合は(1)へ戻る。

【0033】

以上がuDAPLによるRDMA通信手順の概要である。なお、上記はRDMA readの説明であるが、イニシエータが対向ホストへデータを送信するRDMA writeの場合も、データの転送方向が異なる以外は同様の動作手順となる。

【0034】

本明細書は、コネクション設定機構やデータ転送の対象となる主メモリ領域の抽象化機構を持つAPIから成る通信方式を対象とする。現時点でその唯一の具体例として挙げられるものがuDAPLである。

【0035】

【非特許文献1】INFINIBAND NETWORK ARCHITECTURE、MINDSHARE、INC.、Tom Shanley、PC SYSTEM ARCHITECTURE SERIES、Addison-Wesley (2003)、ISBN 0-321-11765-4.

10

20

30

40

50

【非特許文献 2】uDAPL:User Direct Access Programming Library, Version 2.0, DAT collaborative.

【発明の開示】

【発明が解決しようとする課題】

【0036】

バージョン 1. 2 までの従来の u D A P L では、通信経路の耐障害性向上のための仕組み、すなわち、通信経路の冗長化や障害発生時の通信経路の切り替え等の仕組みを提供していなかった。現在の u D A P L バージョン 2. 0 では、これらの機能を H A (High Availability) 機能としてオプションでサポートするとしている（非特許文献 2、Page 58-62）。しかし、その A P I において、E P のコネクション設定の冗長化の指定方法については明記されている（非特許文献 2、Page 231, 233）ものの、L M R と組み合わせた場合の動作等について詳細が明記されておらず、不明な点が多い。

10

【0037】

したがって、現状、相互結合網の耐障害性を図るには、アプリケーションソフトウェアレベルでユーザ自身が明示的に耐障害性を考慮したアーキテクチャを実現する以外に方法がない。その場合、以下の 2 つの自明な方式が考えられる。

【0038】

（方式 1）冗長化した通信経路の全てに常時同じデータを冗長的に流すが、実際にはそのうちの 1 つの通信経路上のデータのみを使用する。障害が発生した際には、使用するデータを他の通信経路上のデータに切り替える。

20

【0039】

（方式 2）通信経路は冗長化するが、ただ 1 つの通信経路にのみデータを流し通信を行う。障害が発生した際には、使用する通信経路を切り替え、その新しい通信経路にのみデータを流す。

【0040】

図 4 は、（方式 1）を図示したものである。ホスト A が外部装置であるディスクからデータを主メモリ内に読み出した後、それがホスト B の主メモリに R D M A によって転送され、ホスト B はその受信データを N I C 経由でネットワークに配信する例である。ホスト A、B 間の R D M A 通信の耐障害性を向上するため、コネクション 1、2 に対応した 2 つの通信経路による冗長化を図っている。

30

【0041】

ホスト A、B の主メモリ上には、通信経路 1、および、通信経路 2 に対応した R D M A 領域 1、2 が存在する。そして、これらの R D M A 領域は複数のセグメントに分割されており、各セグメントに対して 1 つずつ L M R がマッピングされている。R D M A 領域 1 内の各セグメントにマッピングされた L M R の集合を L M R 群 1、R D M A 領域 2 内の各セグメントにマッピングされた L M R の集合を L M R 群 2 と呼ぶ。すなわち、図 4 は、各通信経路に対して図 2 に示した R D M A 領域、および、L M R を複数保持する例である。

【0042】

ホスト A は、主メモリ上のセグメントと同じサイズのデータを読み出し、セグメントへ順番に格納していく。ディスク上の 1 つのデータについて、必ず 2 重に読み出し、一方を R D M A 領域 1 内のセグメント、もう一方を R D M A 領域 2 内のセグメントに格納する。このようにして、同じデータを R D M A 領域 1、2 に冗長化して格納していく。

40

【0043】

また、L M R 群 1、2 を用いて、常時通信経路 1、2 の両方を使用しながら、R D M A 領域 1、2 に格納されたデータがホスト A からホスト B へ転送される（R D M A のイニシエータはどちらのホストでも構わない）。ホスト A の R D M A 領域 1 内のデータは通信経路 1 経由でホスト B の R D M A 領域 1 へ、ホスト A の R D M A 領域 2 内のデータは通信経路 2 経由でホスト B の R D M A 領域 2 へ転送される。

【0044】

各通信経路上では、1 回の R D M A 動作で 1 つのセグメント内のデータが転送され、こ

50

れを繰り返すことによってデータ転送が継続して行われる。なお、このデータ転送動作はディスクからのデータ読み出し動作と並行して行われる。

【0045】

以上のようにして、ディスクから読み出された全てのデータが冗長化され、冗長化された通信経路を経由してホスト間で転送される。

【0046】

ホストBは、ホストAからのデータ受信動作と並行して、コネクション1、2のどちらか一方からのデータをNIC経由でネットワークへ配信する動作を行う。

【0047】

ここで、例えば通信経路1からのデータをネットワークに配信している最中に同通信経路上で障害が発生し通信できなくなった場合、ホストBは、そのNIC上で、ネットワークへ配信するデータを通信経路2からのデータに切り替える。このようにして、ネットワークに対し途切れのないデータ配信を継続する。

【0048】

この例から分かるように、（方式1）では、ホスト間で冗長化された通信経路のうちの一部の通信経路に障害が発生してもホスト間のデータ転送が途切れることがない。図4の例では、ホストBのNIC上でネットワークへ配信するデータの供給元となる通信経路を切り替えるだけで、システムとしての正常な動作が維持できる。データを流す通信経路の切り替えが必要ないので、後述する（方式2）が有するような、通信経路の切り替えに伴うRDMA領域間でのデータのコピーや移動処理が必要ない、という利点がある。

【0049】

しかし、（方式1）では、常時データを冗長化して転送するため、ホストの処理負荷や通信に使用する各種リソースの消費量が増大するという問題がある。図4の例では、

- ・ホストAにおけるディスクからのデータ読み出し
- ・ホストAにおけるデータ送信
- ・ホストBにおけるデータ受信

の処理が常時二重化されている。ホストAの送信処理、および、ホストBの受信処理はRDMAによって行われるため、ホストのCPUにかかる負荷は小さい。しかし、ディスクからのデータ読み出し処理は、ホストの処理負荷を増大させることがある。また、ディスクのような外部装置とのデータの入出力性能は、システムの全体性能の支配項となる場合が多く、I/Oバス等この部分のリソースの消費量を増やすことはシステムの全体性能を向上させる上で得策でないことが多い。特に、高速・大容量のデータ通信を行うシステムほどこのような問題は深刻となる。以上が（方式1）の問題点である。

【0050】

一方、図5は、（方式2）を図示したものである。図4と同様に本例も、ホストAが外部のディスクからデータを主メモリ内に読み出した後、それがホストBの主メモリにRDMAによって転送され、ホストBはその受信データをNIC経由でネットワーク上に配信する。ホストA、B間の通信経路も図4の例と同様に2重化している。

【0051】

図4と異なる点は、2つの通信経路のうちのただ1つの通信経路にのみデータを流して通信を行う点である。正常時において、ホストAのディスクから読み出されたデータは、主メモリの通信経路1用のRDMA領域1へ格納されたのち、RDMAによって通信経路1を用いてデータをホストBへ転送される。ホストBは、主メモリの通信経路1用のRDMA領域1に受信したデータを、NICを経由してネットワークへ配信する。図5ではこの状態でのデータの流れを実線矢印で示している。

【0052】

本動作中に通信経路1上で障害が発生すると、ホストAとホストBは、使用する主メモリのRDMA領域、および、LMR群を通信経路1用から通信経路2用の物へと切り替える。これにより、切り替え後は、図中の破線矢印で示すように、ディスクから読み出されたデータは通信経路2を経由してホストBへ転送され、ネットワークへ配信される。

【0053】

このように（方式2）では、データ転送そのものは冗長化しないため、ホストの処理負荷や各種リソースの消費量が増大するという（方式1）のような問題はない。

【0054】

しかし、（方式2）では、障害発生時にデータを流す通信経路を切り替える操作が必要であり、この時に、主メモリ上のRDMA領域間でデータをコピー（または移動）したり、あるいは、新しく使用するRDMA領域へ外部装置からデータを再度読み出さなければならない場合がある。例えば図6に示すように、通信経路1上で障害が発生した時に、ホストAのRDMA領域1内にまだホストBへの転送が完了していないデータが残っていた場合を考える。この場合ホストAは、このデータを通信経路2経由でホストBへ転送できるようにするため、以下の（A）、（B）のうちのいずれかを実行する必要がある。

10

【0055】

（A）RDMA領域1内の未転送データをRDMA領域2へコピー（または移動）する（図7）。

【0056】

（B）RDMA領域1内の未転送データと同じデータを再度ディスクから読み出し、RDMA領域2へ格納する（図8）。

【0057】

なお、図のように1回のRDMA動作におけるデータ転送単位（本例では1つのセグメント）内のデータのRDMA転送の途中で通信経路1に障害が発生し、当該RDMA転送が途中で異常終了した場合は、上記（A）、（B）のいずれかを完了した後、当該セグメントの先頭からRDMA転送を再開する。これは、各回のRDMAの進行状況はアプリケーションプロセスでは把握できず、異常終了するまでに転送したデータ量を正確に把握することができないため、初めから実行し直す必要があるためである。

20

【0058】

そして、上記（A）または（B）の実行中は、ホストA、B間のデータ転送は停止する。そのため、ホストBからのネットワークへのデータ配信がとぎれないようにするには、ホストA側で上記の動作をできるだけ高速に実行する必要がある。しかし、特にシステムが高速・大容量のデータを扱う場合、上記のコピー（移動）や再読み出しを行うデータ量が大きくなる傾向があり、上記の動作を高速に実行することは容易ではない。

30

【0059】

本発明の目的は、通信経路上で障害が発生した場合の迅速な通信経路の切り替え動作を維持しながら、上述の従来の自明な通信経路冗長化方式が持つ問題を解決することである。

【課題を解決するための手段】

【0060】

上記課題を解決するための手段として、外部装置とのデータの入出力、および、ホスト間でのRDMA転送の対象となる主メモリの領域は単一の物を用意する一方で、冗長化した通信経路毎にその通信経路上で通信を行うためのLMRを用意し、上記単一の主メモリ領域に対して、各通信リンクに対応したLMRを重ねてマッピングする方式を発明した。

40

【0061】

本発明の概要を図9に示す。本発明は、冗長化された通信経路（ここでは2つの通信経路）毎にそれぞれ対応したLMR群を用意する点は「発明が解決しようとする課題」において説明した従来の（方式1）、（方式2）と同様であるが、主メモリ上のRDMA領域は単一の領域を用意する点異なる。

【0062】

正常動作時には、「発明が解決しようとする課題」で述べた従来の（方式2）と同様に、冗長化された複数の通信経路のうちの1つだけを使用してホスト間でのデータ転送を行う。図9の例で言えば、ホストAでは、主メモリ上のセグメントと同じサイズのデータがディスクから読み出されセグメントに格納されていくとともに、LMR群1を用いて通信

50

経路 1 経由で R D M A によりそれらのデータがホスト B へ転送される（R D M A のイニシエータはどちらのホストでもよい）。ホスト B では、L M R 群 1 を用いて通信経路 1 から受信したデータを主メモリに格納するとともに、主メモリ上のデータを N I C 経由でネットワークへ配信する。

【0063】

ここで、図 10 に示すように、通信経路 1 に障害が発生すると、ホスト A、および、ホスト B は R D M A に使用する L M R 群を 1 から 2 へ切り替える。そして、まだホスト B へ転送していない主メモリ上のセグメントから、R D M A によるデータ転送を再開する。

【0064】

同図のように 1 回の R D M A 動作におけるデータ転送単位（本例では 1 つのセグメント）内のデータの R D M A 転送の途中で通信経路 1 に障害が発生し、当該 R D M A 転送が途中で異常終了した場合は、使用する L M R 群を 1 から 2 へ切り替えた後、当該セグメントの先頭から再び R D M A 転送を実行する。これは、各回の R D M A の進行状況はアプリケーションプロセスでは把握できず、異常終了するまでに転送したデータ量を正確に把握することができないため、初めから実行し直す必要があるためである。

【0065】

ホスト A におけるディスクから主メモリへのデータ読み出し、および、ホスト B における主メモリからネットワークへのデータ配信動作は、L M R 群の切り替えの影響を受けずに継続が可能である。

【0066】

本発明では、ディスクからのデータ読み出しは、各データについて 1 度だけ、単一の R D M A 領域に対して行えばよい。また、ネットワークへのデータ配信においても、単一の R D M A 領域から N I C へのデータ読み出しは各データについて 1 度だけ行えばよい。したがって、[発明が解決しようとする課題] で述べた（方式 1）の問題、すなわち、外部装置と主メモリとの間のデータ転送を冗長して行うことに起因する、ホストの処理負荷の増大や各種リソース消費量の増大という問題はない。

【0067】

また、本発明は、障害発生時においてデータを流す通信経路を変更する点では[発明が解決しようとする課題] で述べた（方式 2）と同じである。しかし、本発明では、通信経路の切り替えのために必要な動作は L M R 群の切り替えだけである。図 9 の例では、ホスト A におけるディスクから主メモリへのデータ読み出し、および、ホスト B における主メモリからネットワークへのデータ配信動作は、上記の L M R 群の切り替えの影響を受けずに、単一の R D M A 領域を使用したまま常時継続できる。したがって、[発明が解決しようとする課題] で述べた（方式 2）の問題、すなわち、R D M A 領域間でのデータのコピー（移動）やディスクからのデータの再読み出しのような、高速実行が困難となり得る処理が介在する、という問題もない。

【発明の効果】

【0068】

本発明によれば、通信経路上で障害が発生した場合の迅速な通信経路の切り替え動作を維持しながら、[発明が解決しようとする課題] で述べた従来の自明な通信経路冗長化方式が持つ問題を解決することが可能である。

【発明を実施するための最良の形態】

【0069】

〔発明を実施するための最良の形態 1〕

以下、本発明を実施するための最良の形態 1（以下、形態 1 と記載）について説明する。本形態 1 は、[特許請求の範囲] に記載の請求項 1、3、4 の方法に基づいており、また、それを具現化したサーバ装置であるため請求項 5 にも対応する。本形態 1 の装置全体の構成を図 11 に示す。

【0070】

図 11 に示す相互結合網冗長型マルチメディアサーバ装置が本形態に係るサーバ装置で

10

20

30

40

50

ある。映像データなどの時間連続性を持つマルチメディアデータを外部装置へ配信する機能、および、外部装置から受信したマルチメディアデータを蓄積する機能を有する。外部装置と直接通信する通信サーバモジュール1～M₁、マルチメディアデータが格納される蓄積装置1～M₂、蓄積装置を制御する蓄積サーバモジュール1～M₂、サーバ内データベース、通信サーバモジュールと蓄積サーバモジュール間でのマルチメディアデータ転送に使用されるサーバモジュール間ネットワークスイッチ、および、通信サーバモジュールと蓄積サーバモジュールとサーバ内データベースを接続するローカルエリアネットワークで構成される。

【0071】

通信サーバモジュールと蓄積サーバモジュールは汎用の計算機から成るモジュールで、相互結合網冗長型マルチメディアサーバ装置はこれらをクラスタ化したものである。すなわち、通信サーバモジュールと蓄積サーバモジュールは請求項1に記載の計算機に対応し、相互結合網冗長型マルチメディアサーバ装置は請求項1に記載のサーバ装置に対応する。

10

【0072】

相互結合網冗長型マルチメディアサーバ装置は、端末側ネットワークを介して端末やマルチメディア情報入力装置などの外部装置と接続されている。端末は、同サーバ装置からマルチメディアデータの配信を受ける装置である。また、マルチメディア情報入力装置は、同サーバ装置へ蓄積したいマルチメディアデータを送信する装置である。

【0073】

相互結合網冗長型マルチメディアサーバ装置内のサーバモジュール間ネットワークスイッチは、請求項1に記載の相互結合網に対応する。本形態1では、同スイッチはInfiniBandスイッチで、上位レイヤにuDAPLを用いるものとする。このuDAPLは請求項1に記載の「計算機間のコネクション設定機構および当該計算機の主メモリ領域の抽象化機構を持つ通信方式」に対応する。

20

【0074】

本形態1では、上記スイッチを2つ用いることにより相互結合網を冗長化している。すなわち、請求項1に記載のDの値を2とした場合に対応する。

【0075】

1つのスイッチ、そこから伸びる通信リンク、および、それらの通信リンクによって当該スイッチに接続された全てのHCAからなる1つのサーバモジュール間ネットワークを系と呼び、同図における2つの系をそれぞれ0系、1系とする。また、この0系あるいは1系内において、ある1台の通信サーバモジュールとある1台の蓄積サーバモジュールを接続する通信リンクと同スイッチからなる通信経路を、0系通信経路、1系通信経路、のよう呼ぶ。

30

【0076】

各系の内部では任意のHCA間で通信が可能であるが、系をまたがる通信はできないものとする。

【0077】

各通信サーバモジュールは、端末側ネットワークとの接続のためのネットワークインタフェースカード(NIC)を1枚、および、0系、1系の各サーバモジュール間ネットワークスイッチとの接続のためにInfiniBandホストアダプタカード(HCA)を2枚搭載している。

40

【0078】

各蓄積サーバモジュールは、0系、1系の各サーバモジュール間ネットワークスイッチとの接続のためのInfiniBand HCAを2枚、および、自身が管理する蓄積装置との接続のためのファイバチャネルホストバスアダプタ(HBA)を1枚搭載している。

【0079】

相互結合網冗長型マルチメディアサーバ装置内のローカルエリアネットワークは、通信

50

サーバモジュール、蓄積サーバモジュール、サーバ内データベースを接続するネットワークである。このネットワークは前記 0 系、1 系のサーバモジュール間ネットワークのように計算機間のコネクション設定機構や当該計算機の主メモリ領域の抽象化機構を持つ通信方式を実行可能である必要はなく、L A N (Local Area Network) 等で用いられるイーサネットなどの一般的なネットワークでよい。

【0080】

相互結合網冗長型マルチメディアサーバ装置内のサーバ内データベースは、同サーバ装置に格納されている番組の格納場所や、各蓄積装置の使用状況に関するデータを保持するデータベースである。同サーバ装置の通信サーバモジュール、および、蓄積サーバモジュールはこのローカルエリアネットワークを介してこのサーバ内データベースにアクセスし、上記の情報を得る。このサーバ内データベースを実現する装置としては、前記のデータを保持できるだけの記憶装置を内部に持ち、かつ、通信サーバモジュールや蓄積サーバモジュールとの通信が可能な物であればよく、汎用の計算機等で実現するのが簡明である。このサーバ内データベースは内部に、番組格納情報テーブル、および、蓄積装置使用状況テーブルの 2 つのテーブルを持つ。これらを図 12 に示す。

【0081】

番組格納情報テーブルは、相互結合網冗長型マルチメディアサーバ装置に格納されている各番組のマルチメディアデータがどの蓄積装置に格納されているかを示すテーブルである。図 12 の例では I D が 1 の番組が蓄積装置 5 に、I D が 2 の番組が蓄積装置 M_2 に格納されていることを示している。新しい番組が同サーバ装置に格納されるたびに、テーブル内にこのようなエントリが作成されていく。また、蓄積装置使用状況テーブルは、同サーバ装置の各蓄積装置の空き容量を示すテーブルである。蓄積装置 1 から M_2 の各蓄積装置の空き容量が記憶されている。新しい番組が同サーバ装置に格納されるたびに、テーブル内の空き容量を表す数値が更新されていく。

【0082】

本形態 1 において、マルチメディアデータの配信は以下のように行われる。マルチメディアデータの配信を要求する端末は、サーバ装置内の任意の 1 台の通信サーバモジュールに対し、端末側ネットワークを介してマルチメディアデータの配信要求を発行する。配信要求には、要求元の端末を特定する要求元 I D、および、配信を要求するマルチメディアデータを特定する番組 I D が情報として含まれる（要求元 I D や番組 I D の管理方法についてはここでは規定しない）。

【0083】

この配信要求を受信した通信サーバモジュールは、ローカルエリアネットワークを介してサーバ内データベースにアクセスし、同データベース内の番組格納情報テーブルの当該番組 I D を含むエントリを検索して、当該番組のマルチメディアデータが格納されている蓄積装置番号を得る。

【0084】

そして、サーバモジュール間ネットワークのうち正常に動作している 1 つの系を使用して、上記蓄積装置番号の蓄積装置を管理する蓄積サーバモジュールから当該番組のマルチメディアデータを受信し、これを要求元 I D に相当する端末へ配信する。

【0085】

また、マルチメディアデータの蓄積は以下のように行われる。マルチメディアデータの蓄積を要求するマルチメディア情報入力装置は、サーバ装置内の任意の 1 台の通信サーバモジュールに対し、端末側ネットワークを介してマルチメディアデータの蓄積要求を発行する。蓄積要求には、要求元のマルチメディア情報入力装置を特定する要求元 I D、および、蓄積するマルチメディアデータに新規に割り振る番組 I D が情報として含まれる。

【0086】

蓄積要求を受信した通信サーバモジュールは、ローカルエリアネットワークを介してサーバ内データベースにアクセスし、同データベース内の蓄積装置使用状況テーブルを参照して、空き容量が最も多い蓄積装置を当該マルチメディアデータの格納先蓄積装置に決定

する。

【0087】

そして、サーバモジュール間ネットワークのうち正常に動作している1つの系を使用して、上記蓄積装置を管理する蓄積サーバモジュールに対し、マルチメディア情報入力装置から受信したマルチメディアデータを送信する。

【0088】

蓄積サーバモジュールは、受信したマルチメディア情報を自身が管理する蓄積装置へ格納する。当該番組の全てのマルチメディア情報の蓄積装置への格納が完了すると、蓄積サーバモジュールは、サーバ内データベースにアクセスし、当該番組IDとそれを格納した蓄積装置番号からなる新しいエントリを番組格納情報テーブルへ追加し、さらに、番組格納後の当該蓄積装置の空き容量を蓄積装置使用状況テーブルに記録し、処理を終了する。

10

【0089】

なお、1番組のマルチメディア情報は固定長のセグメントに分割されて蓄積装置に格納されるものとする。蓄積装置からの読み出しや蓄積装置への書き込み、通信サーバモジュールと蓄積サーバモジュール間の転送は、いずれもこのセグメントを単位に行う。

【0090】

また、通信サーバモジュールと蓄積サーバモジュールとの間のデータ転送はUDPのAPIを介したRDMAにより実行する。1回のRDMA操作で上記セグメントの1つが両者の間で転送される。この転送動作を繰り返すことにより当該番組のマルチメディアデータの転送が行われる。なお、本形態1においては、通信サーバモジュールがRDMAのイニシエータであるものとする。

20

【0091】

なお、端末側ネットワークとのデータの送受信時の形式については特に規定しない。

【0092】

以上が本形態1の装置構成、および、動作の概要の説明である。これらを踏まえ、以下に本発明技術を含んだ詳細な説明を行う。まず、図13に通信サーバモジュールの内部構造の詳細を示す。

【0093】

RDMA領域は、番組のマルチメディア情報を格納するためのバッファで、オペレーティングシステムが提供する仕組みを用いて主メモリに確保した領域である。N個の領域に分かれており、各領域にマルチメディア情報の1つのセグメントを格納する。

30

【0094】

配信動作の場合はRDMA転送によって蓄積サーバモジュールから受信したセグメントが格納され、端末側ネットワークへ配信されるまでのバッファとして機能する。

【0095】

蓄積動作の場合は端末側ネットワークから受信したマルチメディア情報が格納され、RDMA転送によってセグメント単位で蓄積サーバモジュールへ転送されるまでのバッファとして機能する。

【0096】

読み出しポインタ(`rd_ptr`)、書き込みポインタ(`wr_ptr`)、フルフラグ(`full_flag`)、エンプティフラグ(`empty_flag`)は、上記RDMA領域へのセグメントの格納状況を管理するための、ソフトウェアプログラム変数である。

40

【0097】

`rd_ptr`は、RDMA領域に既に格納されているセグメントを読み出す際に、読み出し元となるバッファ番号を保持するポインタである。配信動作の場合は、NIC経由で端末側ネットワークへ配信するセグメントを指し、蓄積動作の場合は、RDMAで蓄積サーバモジュールへ転送するセグメントを指す。

【0098】

`wr_ptr`は、RDMA領域へセグメントを書き込む際に、書き込み先となるバッファ番号を保持するポインタである。配信動作の場合は、蓄積サーバモジュールからRDMA

50

Aで転送されてきたセグメントの格納先バッファを指し、蓄積動作の場合は、端末側ネットワークから受信したマルチメディアデータの格納先バッファを指す。

【0099】

`full_flag`は、全てのバッファにセグメントが格納されていれば値1、それ以外の場合は値0を保持するフラグである。

【0100】

`empty_flag`は、どのバッファにもセグメントが格納されていない場合は値1、それ以外の場合は値0を保持するフラグである。

【0101】

これらのポインタとフラグを用いて、N個のバッファからなる有限長のRDMA領域を仮想的なリングバッファとして使用することにより、セグメント長がNよりも長い番組データを扱うことができる。

【0102】

要求送信バッファは、RDMA領域と同様の方法で主メモリ上に確保した領域で、RDMA転送の実行前に蓄積サーバモジュールに対して送信するメモリ情報要求を格納するバッファである。

【0103】

メモリ情報受信バッファは、上記と同様の方法で主メモリ上に確保した領域で、上記メモリ情報要求に対して蓄積サーバモジュールから送られてきたメモリ情報を受信するためのバッファである。

【0104】

上記RDMA領域、要求送信バッファ、メモリ情報受信バッファを全て合わせたものが請求項1に記載のデータ格納手段に対応する。したがって、これらは全て、冗長化した通信経路の数には関係なく1つだけ確保する。

【0105】

データ転送用LMR群0、1はuDAPLのオブジェクトであり、それぞれ、冗長化した2つの0系、1系通信経路用のLMRである。図9のLMR群1、2に相当する物である。主メモリ上のRDMA領域の各バッファ[i] ($0 \leq i \leq N-1$)に対して、0系用のLMR[0][i]と1系用のLMR[1][i]の2つのLMRがマッピングされている。

【0106】

本形態1では、マルチメディアデータの転送だけでなくRDMAに伴うメモリ情報の送受信においても2つの通信経路を用いて信頼性の向上を図る。そのために、上記要求送信バッファ、および、メモリ情報受信バッファについても、それらにマッピングされた0系、1系通信経路用のLMRを用意する。S-LMRが要求送信バッファ、R-LMRがメモリ情報受信バッファに対応したLMRである。

【0107】

1つのデータ転送用LMR群と1つのメモリ情報交換用LMR群を合わせた物が請求項1に記載のメモリ領域抽象化手段に対応し、本形態1では2つのメモリ領域抽象化手段を保持した場合に対応する。

【0108】

`request_EVD`は、蓄積サーバモジュールへ送信するメモリ情報要求の送信完了、および、1回のRDMA `read/write`動作の完了を通知するuDAPLオブジェクトである。0系、1系通信経路毎に用意する。

【0109】

`receive_EVD`は、蓄積サーバモジュールからのメモリ情報の受信完了を通知するuDAPLオブジェクトである。0系、1系通信経路毎に用意する。

【0110】

EPは、蓄積サーバモジュールとの通信を行うために確立するコネクションの端点となるuDAPLオブジェクトである。本形態1では、0系、1系の2つの通信経路上で通信

10

20

30

40

50

サーバモジュールと蓄積サーバモジュール間のコネクションを確立することでコネクションを冗長化し、両者間の通信の高信頼化を図る。そのために0系、1系用の2つのEPを用いる。すなわち、本形態1は、合計2本の冗長な通信経路を保有する請求項1に記載の経路冗長化手段の一例を実現している。

【0111】

各系のEPは、その生成時において、それぞれ図中のrequest EVD、receive EVDと関連づけられているものとする。これにより、ある系のEPを経由して実行する動作の完了通知は同じ系内の当該EVDを通して行われる。

【0112】

本形態1では、LMR群とEPについては、同じ系内の物どうしの組み合わせでしか使用しない。このため、各系のLMR群とEPはその系内の1つのPZに関連づけて生成し、異なる系に属するこれらのリソースを混在させた処理が行えないようにしている。

10

【0113】

通信経路監視部は、通信サーバモジュールと蓄積サーバモジュールを結ぶ各通信経路の通信可否状態を監視するソフトウェアモジュールである。0系、1系の各通信経路に対応して1つずつ存在する。また、その内部に、当該通信経路の状態を記録するためのソフトウェアプログラム変数であるステータス変数を持つ。請求項4に記載の通信経路障害・回復検出手段に対応する。

【0114】

各系の通信経路監視部は、当該通信経路の対向の蓄積サーバモジュール上の通信経路監視部と相互に定期的にメッセージ交換を行い、事前に設定した規定時間（請求項3に記載の「メッセージ交換の規定時間」に対応するもの）に基づいて、当該通信経路における障害発生や障害からの回復を検出する。

20

【0115】

メッセージ交換が上記規定時間以内に終了できなかった場合、当該通信経路に障害が発生して使用不可能になったとみなし、ステータス変数に値「断」を設定する。メッセージ交換が上記規定時間以内に終了できた場合は、当該通信経路が使用可能であるとみなし、ステータス変数の値を「正常」に設定する。

【0116】

なお、このメッセージ交換は、uDAPLのAPIで提供されるメッセージsend/receiveやRDMAではなく、ICMP echo request/replyのような軽量な通信を使用する。

30

【0117】

uDAPLには通信経路の障害をアプリケーションソフトウェアに対して割り込み通知する機能がない。しかし、後述するように、EVDを経由して各種の処理の完了通知を待つ際、待ち時間の規定時間（請求項3に記載の「データ転送の規定時間」に対応するもの）を設定することができる。したがって、データ転送の際にこの規定時間を用いて通信経路の障害を検出することも可能である。

【0118】

これに対し、本形態1は、請求項3、4に記載の方法、すなわち、データ転送の規定時間と通信経路監視部の2つを併用して通信経路の状態を判断する方法を用いている。これは以下の2点において前者より優れている。

40

【0119】

・マルチメディアデータの1つのセグメントのデータサイズは非常に大きく、その転送にかかる時間のばらつき（ジッタ）も大きくなる傾向がある。したがって、データ転送の規定時間を元に通信経路の障害検出を行う方法では、このジッタを吸収できるだけに十分に長い規定時間を設定しなければ、データ転送時の単なる処理遅れと通信経路障害を混同する可能性が生じる。しかし、これでは迅速な障害検出が行えない。請求項3、4に記載の方法では、上記のような誤判断なく、しかも、障害検出の迅速化が図れる。

【0120】

50

・データ転送動作とは独立に動作する通信可否状態の監視機構により、障害検出だけでなく障害からの回復を検出できる。

【0121】

以上が通信サーバモジュールの内部構造の詳細説明である。次に、図14に蓄積サーバモジュールの内部構造の詳細を示す。

【0122】

蓄積サーバモジュールの0系EPと通信サーバモジュールの0系EPとの間、および、蓄積サーバモジュールの1系EPと通信サーバモジュールの1系EPとの間でコネクションが確立されている。

【0123】

主メモリ上のRDMA領域は、番組のマルチメディア情報を格納するためのバッファで、配信動作の場合は蓄積装置から読み出されたセグメントが格納され、RDMAによって通信サーバモジュールへ送信するまでのバッファとして機能する。蓄積動作の場合はRDMA転送によって通信サーバモジュールから受信したセグメントが格納され、蓄積装置へ書き込むまでのバッファとして機能する。

【0124】

本RDMA領域も通信サーバモジュールにおけるそれと同様に、rd_ptr、wr_ptr、full_flag、empty_flagを用いて仮想的なリングバッファとして使用する。

【0125】

メモリ情報送信バッファは、通信サーバモジュールからのメモリ情報要求に対して送信するメモリ情報を格納するバッファである。

【0126】

要求受信バッファは、通信サーバモジュールからのメモリ情報要求を受信するためのバッファである。

【0127】

これら以外の構成要素については、通信サーバモジュールの対応する構成要素と同じあるため説明は省略する。

【0128】

以上が通信サーバモジュール、および、蓄積サーバモジュールの内部構造の詳細説明である。以上のように構成した相互結合網冗長型マルチメディアサーバ装置の動作の詳細を以下に説明する。

【0129】

なお、以下では、同サーバ装置から端末へのマルチメディアデータの配信動作だけを説明する。蓄積動作の場合はマルチメディアデータの転送方向が逆になるだけで、本質的な差異は存在しない。

【0130】

また、通信／蓄積サーバモジュール内の各構成要素を生成するタイミングについては本形態1では規定せず、全ての動作の開始前に各構成要素が存在していることを前提とする。したがって、図13と図14に示した通信サーバモジュール／蓄積サーバモジュール間のコネクションについても、以下の説明の配信動作における対向のサーバモジュールどうしを接続したもので、事前の存在を前提とする。

【0131】

また、簡略化のため、図13、図14においてはEP間コネクション確立通知用EVDの記載を、図14においてはデータ転送用LMRに対応するRMRの記載は省略している。

【0132】

まず、図15に、本形態1における通信サーバモジュールの処理を示すフローチャートを示す。同図に示すS1～S15に従って通信サーバモジュールの処理を説明する。

【0133】

(S1) 端末から番組の配信要求を受信する。

【0134】

(S2) サーバ内データベースにアクセスし、上記配信要求に含まれる番組IDをキーとして同データベース内の番組格納情報テーブルを検索し、当該番組が格納されている蓄積装置を管理する蓄積装置番号を得る。

【0135】

(S3) S2で得た蓄積装置番号の蓄積装置を管理する蓄積サーバモジュールに対し、当該番組IDを含む配信開始通知を発行する。

【0136】

(S4) 当該蓄積サーバモジュールからの前処理完了通知を受信するまで待つ。この前処理完了通知は、蓄積サーバモジュール側が各種リソースの初期化を完了し、当該番組のセグメントの転送が可能となったことを示す。

【0137】

(S5) 端末への初期配信時刻、すなわち、当該番組の当該端末への配送を開始する時刻を設定する。例えば、初期配送時刻を、端末からの配信要求を受信した時刻に設定すると、蓄積サーバモジュールから当該番組の最初のセグメントを受信した時に配信が開始される。

【0138】

(S6) `rd_ptr`、および、`wr_ptr`の値をそれぞれ、主メモリ内のバッファ[0]を指すよう、0に初期化する。また、上記バッファがまだ空（セグメントが一切格納されていない）であるため、`full_flag`を0、`empty_flag`を1に初期化する。

【0139】

(S7) 冗長化された2つの通信経路のうち、使用する通信経路の系番号を表す変数Iを0に初期化する（最初は0系通信経路から使用する）。

【0140】

(S8) 後述する「端末へのセグメント配信処理」を起動する。

【0141】

(S9) 蓄積サーバモジュールに対して送信するメモリ情報要求の内容を主メモリ上の要求送信バッファへ書き込む。このメモリ情報要求は、転送対象となるセグメントを特定する番号等を含めることによりRDMA read動作の対象となる蓄積サーバモジュールのLMRを特定するものであるが、本明細書ではその形式は規定しない。

【0142】

(S10) `full_flag`の値が0になるまで待つ。すなわち、主メモリ上にバッファの空きがあり、蓄積サーバモジュールからのセグメントの受信が可能になるまで待つ。

【0143】

(S11) 蓄積サーバモジュールから当該番組の1つのセグメントを受信する。本ステップの詳細は後述する。

【0144】

(S12) 1つのセグメントを受信したら、`wr_ptr`がその次のセグメントの格納先のバッファを指すよう、値を更新する。すでに最後尾のバッファ[N1]を指していれば先頭バッファ[0]を、そうでなければ、1つ先のバッファを指すように値を更新する。

【0145】

(S13) セグメントの受信直後は主メモリ上の全バッファが空でないことが事実であるため、`empty_flag`の値を0にセットする。

【0146】

(S14) S12で`wr_ptr`の値を更新した結果、もし`rd_ptr`の値と等しくなれば、主メモリ上の全バッファがセグメントデータによって占有されたことを意味する

10

20

30

40

50

ため、`full_flag`の値を1にセットする。そうでなければ何もしない。

【0147】

(S15) S11で蓄積サーバモジュールから受信したセグメントが当該番組の最終セグメントであれば、処理を終了する。そうでなければS9へ戻る。

【0148】

図16は、図15のS8において起動された「端末へのセグメント配信処理」のフローチャートである。同図に示すS20～S27に従って本処理を説明する。

【0149】

(S20) 現在の時刻が、設定されている「端末への配信時刻」を過ぎるまで待つ。

【0150】

(S21) 主メモリの`empty_flag`を参照し、その値が0、すなわち、配信すべきセグメントがあることを確認する。`empty_flag`の値が1であれば0になるまで待つ。

【0151】

(S22) バッファ`[rd_ptr]`に格納されているセグメントを、S1で受信した配信要求内の要求元IDで特定される端末へ配信する。本明細書では、端末側ネットワークへ配信する際のデータ形式については規定しない。

【0152】

(S23) 1つのセグメントを配信したら、`rd_ptr`がその次のセグメントの格納先のバッファを指すように、値を更新する。すでに最後尾のバッファ`[N1]`を指していれば先頭バッファ`[0]`を、そうでなければ、1つ先のバッファを指すように値を更新する。

【0153】

(S24) セグメントの配信直後は、主メモリのバッファのうちセグメントが格納されていない空のバッファが少なくとも1つはあるので、`full_flag`の値を0にセットする。

【0154】

(S25) S23で`rd_ptr`の値を更新した結果、もし`wr_ptr`の値と等しくなれば、主メモリ上の全バッファが空となったことを意味するため、`empty_flag`の値を1にセットする。そうでなければ何もしない。

【0155】

(S26) 現在設定されている「端末への配信時刻」に、端末での1セグメントの再生時間を加えたものを、次の配信時刻として設定する。

【0156】

(S27) S22で配信したセグメントが当該番組の最終セグメントであれば、処理を終了する。そうでなければS20へ戻る。

【0157】

図17は、図15のS11「蓄積サーバモジュールからのセグメント受信処理」を詳細に示したものである。同図に示すS30～S44に従って本処理を説明する。

【0158】

(S30) 要求送信バッファ内の内容をメモリ情報要求として、S-LMR`[I]`を用いてI系通信経路を介して蓄積サーバモジュールへ送信する。このメモリ情報要求はRDMAではなく、`UDP`で用意されている単純なメッセージ`send`機構を用いる。

【0159】

(S31) S30のメッセージ`send`が事前に設定した規定時間内に正常に終了すれば、S35へ移る。規定時間内に終了しなければS32へ移る。

【0160】

(S32) I系通信経路のステータス変数の値を確認する。その値が「正常」なら、メッセージ`send`の処理遅れの可能性があるため、S31へ戻る。「正常」でなければ(「断」であれば)S33へ移る。

10

20

30

40

50

【0161】

(S33) I系通信経路に障害が発生したと判断し、代わりに使用できる通信経路が他にあるか否かを判断する。なければここで処理を終了(異常終了)する。あればS34へ移る。

【0162】

(S34) 新しく使用する通信経路の系番号を決定する。ステータスが「正常」な通信経路の系番号のうち、最小の番号を変数Iにセットする。その後、S30に戻り、新しい通信経路上でメモリ情報要求を送信し直す。

【0163】

(S35) R-LMR[I]を用いてI系通信経路を介してメモリ情報を蓄積サーバモジュールから受信し、内容をメモリ情報受信バッファに格納する。このメモリ情報の受信はRDMAではなく、uDAPLで用意されている単純なメッセージreceive機構を用いる。

10

【0164】

(S36) S35のメッセージreceiveが事前に設定した規定時間内に正常に終了すれば、S40へ移る。規定時間内に終了しなければS37へ移る。

【0165】

(S37) I系通信リンクのステータス変数の値を確認する。その値が「正常」なら、メッセージreceiveの処理遅れの可能性があるため、S36へ戻る。「正常」でなければ(「断」であれば)S38へ移る。

20

【0166】

(S38) I系通信経路に障害が発生したと判断し、代わりに使用できる通信経路が他にあるか否かを判断する。なければここで処理を終了(異常終了)する。あればS39へ移る。

【0167】

(S39) 新しく使用する通信経路の系番号を決定する。ステータスが「正常」な通信経路の系番号のうち、最小の番号を変数Iにセットする。その後、S35に戻り、新しい通信リンク上でメモリ情報を受信し直す。

【0168】

(S40) 蓄積サーバモジュールから番組情報のセグメントを受信するために、LMR[I][rd_ptr]を受信用LMRに指定し、S36で受信したメモリ情報で指定された蓄積サーバモジュールのバッファのアドレスをターゲットとするRDMA readを発行する。

30

【0169】

(S41) S40のRDMA readが事前に設定した規定時間内に正常に終了しセグメントをバッファ[I][rd_ptr]に受信できれば、図15のS12へ移る。規定時間内に終了しなければS42へ移る。

【0170】

(S42) I系通信経路のステータス変数の値を確認する。その値が「正常」なら、RDMA readの処理遅れの可能性があるため、S41へ戻る。「正常」でなければ(「断」であれば)S43へ移る。

40

【0171】

(S43) I系通信経路に障害が発生したと判断し、代わりに使用できる通信経路が他にあるか否かを判断する。なければここで処理を終了(異常終了)する。あればS44へ移る。

【0172】

(S44) 新しく使用する通信経路の系番号を決定する。ステータスが「正常」な通信経路の系番号のうち、最小の番号を変数Iにセットする。その後、S40に戻り、新しい通信経路上でRDMA readをやり直す。

【0173】

50

図 18 は、図 13 における 0 系、1 系通信経路監視部の処理を示したフローチャートである。各系の通信経路監視部は互いに独立に、また、図 15、図 16、図 17 に示した通信サーバモジュールの動作とも独立に、それぞれ同図のように動作する。同図の S50～S60 に従って通信経路監視部の処理を説明する。

【0174】

(S50) 自通信経路の状態を表すステータス変数の値を「正常」に初期化する。

【0175】

(S51) 対向ホスト上の、同じ系に対応する通信経路監視部に対し、ICMP echo リクエストパケットを送信する。

【0176】

(S52) S51 の対向の通信経路監視部から、事前に設定した規定時間内に ICMP echo リプライパケットを受信できれば、自通信経路に問題はないと判断し、S51 へ戻って監視を継続する。受信できなければ S53 へ移る。

【0177】

(S53) 自通信経路に障害が発生したと判断し、ステータス変数の値を「断」に変更する。

【0178】

(S54) 自通信経路の回復を検出するため、S51 の対向の通信経路監視部に対し、ICMP echo リクエストパケットを送信する。

【0179】

(S55) S54 の対向の通信経路監視部から、事前に設定した規定時間内に ICMP echo リプライパケットを受信できれば、自通信経路が回復したと判断し、S56 へ移る。受信できなければ S54 へ戻って自通信経路の回復検出動作を継続する。

【0180】

(S56) 自通信経路の EP と対向ホスト上の EP との間の接続を切断する。

【0181】

(S57) 自通信経路の EP を消去する。

【0182】

(S58) 自通信経路の新しい EP を生成する。

【0183】

(S59) S58 で生成した新しい EP を用いて、自通信経路における対向の EP と新規に接続を確立する。

【0184】

(S60) 自通信経路のステータス変数の値を「正常」に戻し、S51 へ移って通常の回線監視動作を行う。

【0185】

InfinitiBand には、通信経路に対して一度発行した通信処理を明示的に取り消すための機能がない。そのため、障害から回復した通信経路をそのまま再使用せずに EP と接続を消去した後に新しいものを作り直す処理をステップ 56 からステップ 59 でおこなっている。これにより、通信経路に障害が発生した瞬間に実行中であった通信処理が障害からの回復後に誤って再起動されることを防止する。

【0186】

図 19 は、本形態 1 における蓄積サーバモジュールの処理を示すフローチャートである。同図の S70～S81 に従って、通信サーバモジュールの処理を説明する。

【0187】

(S70) 通信サーバモジュールが図 15 の S3 において送信した配信開始通知を受信する。

【0188】

(S71) rd_ptr、および、wr_ptr の値をそれぞれ、主メモリ内のバッファ [0] を指すよう、0 に初期化する。また、上記バッファがまだ空（セグメントが一切

10

20

30

40

50

格納されていない)であるため、`full_flag`を0、`empty_flag`を1に初期化する。

【0189】

(S72) 冗長化された2つの通信リンクのうち、使用する方の通信経路を示す変数Iを0に初期化する(最初は0系通信リンクから使用する)。

【0190】

(S73) 後述する「蓄積装置からのセグメント読み出し処理」を起動する。

【0191】

(S74) `empty_flag`の値が0になるまで待つ。すなわち、蓄積装置から読み出され主メモリ上のバッファに格納されたセグメントが存在し、通信サーバモジュールへのセグメントの送信が可能になるまで待つ。

10

【0192】

(S75) バッファ[I][`rd_ptr`]内のセグメントが当該番組の最初のセグメントであればS76へ、そうでなければS77へ移る。

【0193】

(S76) 通信サーバモジュールに対して前処理完了通知を発行する。

【0194】

(S77) 通信サーバモジュールへ当該番組の1つのセグメントを送信する。本ステップの詳細は後述する。

【0195】

20

(S78) 1つのセグメントを送信したら、`rd_ptr`がその次に送信するセグメントの格納先のバッファを指すよう、値を更新する。すでに最後尾のバッファ[N1]を指していれば先頭バッファ[0]を、そうでなければ、1つ先のバッファを指すように値を更新する。

【0196】

(S79) セグメントの送信直後は主メモリ上の全バッファが満杯ないことが確実にあるため、`full_flag`の値を0にセットする。

【0197】

(S80) S78で`rd_ptr`の値を更新した結果、もし`wr_ptr`の値と等しくなれば、主メモリ上の全バッファが空となったことを意味するため、`empty_flag`の値を1にセットする。そうでなければ何もしない。

30

【0198】

(S81) S77で通信サーバモジュールへ送信したセグメントが当該番組の最終セグメントであれば、処理を終了する。そうでなければS74へ戻る。

【0199】

図20は、図19のS73において起動された「蓄積装置からのセグメント読み出し処理」の処理を示すフローチャートである。同図のS90～S95に従って、本処理を説明する。

【0200】

(S90) `full_flag`の値が0になるまで待つ。すなわち、主メモリ上にバッファの空きがあり、蓄積装置から読み出すセグメントの格納が可能になるまで待つ。

40

【0201】

(S91) 蓄積装置から当該番組のセグメントを1つ読み出し、主メモリ上のバッファ[`wr_ptr`]に格納する。

【0202】

(S92) 1つのセグメントをバッファに格納したら、`wr_ptr`がその次に蓄積装置から読み出したセグメントの格納先のバッファを指すよう、値を更新する。すでに最後尾のバッファ[N1]を指していれば先頭バッファ[0]を、そうでなければ、1つ先のバッファを指すように値を更新する。

【0203】

50

(S 9 3) 蓄積装置から読み出したセグメントを主メモリ上のバッファへ格納した直後は、全バッファが空でないことが確実であるため、empty_flag の値を 0 にセットする。

【0204】

(S 9 4) S 9 3 で wr_ptr の値を更新した結果、もし rd_ptr の値と等しくなれば、主メモリ上の全バッファがセグメントデータによって占有されたことを意味するため、full_flag の値を 1 にセットする。そうでなければ何もしない。

【0205】

(S 9 5) S 9 1 で蓄積装置から読み出したセグメントが当該番組の最終セグメントであれば、処理を終了する。そうでなければ S 9 0 へ戻る。

10

【0206】

図 2 1 は、図 1 9 の S 7 7 「通信サーバモジュールへのセグメント送信処理」を詳細に示したものである。同図に示す S 1 0 0 ~ S 1 1 1 に従って本処理を説明する。

【0207】

(S 1 0 0) R-LMR [I] を用いて I 系通信経路を介してメモリ情報要求を通信サーバモジュールから受信し、内容を要求受信バッファに格納する。このメモリ情報要求の受信は RDMA ではなく、uDAPL で用意されている単純なメッセージ receive 機構を用いる。

【0208】

(S 1 0 1) S 1 0 0 のメッセージ receive が事前に設定した規定時間内に正常に終了すれば、S 1 0 5 へ移る。規定時間内に終了しなければ S 1 0 2 へ移る。

20

【0209】

(S 1 0 2) I 系通信経路のステータス変数の値を確認する。その値が「正常」なら、メッセージ receive の処理遅れの可能性があるため、S 1 0 1 へ戻る。「正常」でなければ（「断」であれば）S 1 0 3 へ移る。

【0210】

(S 1 0 3) I 系通信経路に障害が発生したと判断し、代わりに使用できる通信経路が他にあるか否かを判断する。なければここで処理を終了（異常終了）する。あれば S 1 0 4 へ移る。

【0211】

30

(S 1 0 4) 新しく使用する通信経路の系番号を決定する。ステータスが「正常」な通信経路の系番号のうち、最小の番号を変数 I にセットする。その後、S 1 0 0 に戻り、新しい通信経路上でメモリ情報を受信し直す。

【0212】

(S 1 0 5) 通信サーバモジュールへ送信するメモリ情報の内容をメモリ情報送信バッファへ書き込む。メモリ情報には、バッファ [rd_ptr] に対応する rmr_triplet（開始アドレス、レンジス、アクセスキー）が含まれる。

【0213】

(S 1 0 6) 主メモリ上のバッファ [rd_ptr] に蓄積装置から読み出したセグメントが格納された状態になるまで待つ。

40

【0214】

(S 1 0 7) S-LMR [I] を用いて I 系通信経路を介して、バッファ [rd_ptr] に関するメモリ情報を通信サーバモジュールへ送信する。このメモリ情報の送信は RDMA ではなく、uDAPL で用意されている単純なメッセージ send 機構を用いる。

【0215】

(S 1 0 8) S 1 0 6 のメッセージ send が事前に設定した規定時間内に正常に終了すれば、図 1 9 の S 7 8 へ移る。終了しなければ S 1 0 8 へ移る。

【0216】

(S 1 0 9) I 系通信経路のステータス変数の値を確認する。その値が「正常」なら、メッセージ send の処理遅れの可能性があるため、S 1 0 7 へ戻る。「正常」でなければ

50

ば（「断」であれば）S 1 0 9 へ移る。

【0 2 1 7】

（S 1 1 0）I 系通信経路に障害が発生したと判断し、代わりに使用できる通信経路が他にあるか否かを判断する。なければここで処理を終了（異常終了）する。あれば S 1 1 0 へ移る。

【0 2 1 8】

（S 1 1 1）新しく使用する通信経路の系番号を決定する。ステータスが「正常」な通信経路の系番号のうち、最小の番号を変数 I にセットする。その後、S 1 0 6 に戻り、新しい通信リンク上でメモリ情報を送信し直す。

【0 2 1 9】

なお、図 1 4 における 0 系、1 系通信経路監視部は、通信サーバモジュールのそれらと同様に、それぞれが独立に、また、図 1 9、図 2 0、図 2 1 に示した蓄積サーバモジュールの動作とも独立に動作する。その動作は、図 1 8 に示した通信サーバモジュールの通信経路監視部の動作と同じである。

【0 2 2 0】

本形態 1 は 2 個のサーバモジュール間ネットワークスイッチを用いているので 2 本のコネクションを備えているが、 $D (\geq 2)$ 個のサーバモジュール間ネットワークスイッチを用いる場合は、D 本のコネクションを備え、データ転送用 L M R 群等もこれに対応して各モジュールで D 個ずつ備えることはいうまでもない。

【0 2 2 1】

〔発明を実施するための最良の形態 2〕

以下、本発明を実施するための最良の形態 2（以下、形態 2 と記載）について説明する。本形態 2 は、〔特許請求の範囲〕に記載の請求項 2、3、4 の方法に基づいており、また、それを具現化したサーバ装置であるため請求項 6 にも対応する。本形態 2 のサーバ装置全体の構成を図 2 2 に示す。

【0 2 2 2】

図 1 1 に示した形態 1 との違いは、0 系－1 系間接続用通信リンクが存在する点である。これは請求項 2 に記載の相互結合網間通信リンクに対応するものである。この通信リンクにより、本形態 2 では、0 系 H C A と 1 系 H C A との間の通信が可能となっている。

【0 2 2 3】

図 2 2 における通信サーバモジュールと蓄積サーバモジュールの内部構造の詳細を図 2 3 に示す。

【0 2 2 4】

通信サーバモジュール上のデータ格納手段は、請求項 1 に記載のそれに対応するもので、図 1 3 のバッファ、要求送信バッファ、および、メモリ情報受信バッファを合わせて簡略化して記載したものである。また、メモリ領域抽象化手段 1～4 は、請求項 1 に記載のメモリ領域抽象化手段に対応するもので、図 1 3 のデータ転送 L M R 群とメモリ情報交換用 L M R 群を合わせて簡略化して記載したものである。通信経路監視部 1～4 は、請求項 4 に記載の通信経路障害・回復検出手段に対応するものである。また、図 1 3 と同様に各コネクションに対応して P Z と各 E V D を備える記載を省略している。また、図 1 3 と同様に読み出しポインタ、書き込みポインタ、フルフラグ、エンプティフラグも備えるが記載を省略している。

【0 2 2 5】

蓄積サーバモジュール上のデータ格納手段は、請求項 1 に記載のそれに対応するもので、図 1 4 のバッファ、メモリ情報送信バッファ、および、要求受信バッファを合わせて簡略化して記載したものである。その他の部分については、通信サーバモジュールの場合と同じように図 1 4 の一部を省略して記載している。

【0 2 2 6】

形態 1 との違いは、通信サーバモジュールと蓄積サーバモジュールの 0 系通信リンクと 1 系通信リンクをまたがるものを含む合計 4 本のコネクションを備え、メモリ領域抽象化

10

20

30

40

50

手段、E P、通信経路監視部、および、記載を省略したP Z、各E V Dもこの4本のコネクションに対応して両計算機で4つずつ備えている点である。

【0227】

これにより、計算機間通信の耐障害性を形態1の場合よりも高めている。形態1では、例えば通信サーバモジュールの0系通信リンクと蓄積サーバモジュールの1系通信リンクの合計2箇所が切断した場合、両計算機間の通信ができなくなる。本形態2ではこのような場合でも、通信サーバモジュールの1系通信リンクと蓄積サーバモジュールの0系通信リンクを使用したコネクション3を使用して通信を継続することが可能である。

【0228】

サーバ内データベースについても、形態1と全く同様のものであり、図12に示したものと同様の情報を内部に持つ。

【0229】

図23の各部の動作フローの詳細については、形態1とほぼ同様のため、記載を省略する。

【0230】

なお、本形態2においては、コネクション1に相当する通信経路を通信経路1、コネクション2に相当する通信経路を通信経路2、以下同様に、通信経路3、4のように呼ぶ。

【0231】

また、上記の各通信経路に対応するリソースであるコネクション、E P、メモリ領域抽象化手段、通信経路監視部への番号の付与について、R D M Aのイニシエータである通信サーバモジュール側を基準にして図23のように1から4の番号を付与し、蓄積サーバモジュール側では、通信サーバモジュール側の対応するリソースの番号と同じ番号を付与する。これにより、通信経路の切り替え処理（形態1における図17、および、図21に含まれる処理）において、形態1の図17と図21と同じ方法で通信サーバモジュールと蓄積サーバモジュールの間で使用する通信経路を常時同期させることができる。

【0232】

本形態2は2本の通信リンクを有するものであるので4本のコネクションを備えているが、 $D \geq 2$ 本の通信リンクを有する場合は、 D^2 本のコネクションを備え、メモリ領域抽象化手段等もこれに対応して各計算機で D^2 個ずつ備えることはいうまでもない。

【0233】

以上の本発明の各形態の説明で明らかなように、本発明の相互結合網冗長型マルチメディアサーバ装置は次のような利点がある。

【0234】

まず、同サーバ装置から番組のマルチメディア情報を配信する場合、マルチメディア情報のセグメントの読み出しは各セグメントについて1度だけ、単一のR D M A領域に対して行えばよい。また、ネットワークへのデータ配信においても、単一のR D M A領域からN I Cへのデータ読み出しは各データについて1度だけ行えばよい。

【0235】

したがって、[発明が解決しようとする課題]で述べた（方式1）の問題、すなわち、外部装置と主メモリとの間のデータ転送を冗長して行うことに起因する、ホストの処理負荷の増大や各種リソース消費量の増大という問題を回避することができる。これは、同サーバ装置へ番組のマルチメディア情報を蓄積する場合にも言える。

【0236】

さらに、本発明において、通信リンクの障害発生時にデータを流す通信リンクを変更する点では[発明が解決しようとする課題]で述べた（方式2）と同じである。しかし、本発明では通信リンクの切り替えのために必要な動作はL M R群の切り替えだけである。そして、この切り替え処理にはほとんど負荷がかからず、瞬時に終了する。

【0237】

したがって、同サーバ装置から番組のマルチメディアデータを配信する場合、ホストBにおける主メモリからネットワークへのデータ配信動作は上記のL M R群の切り替えの影

10

20

30

40

50

響をほとんど受けずに、とぎれることなく正常に継続できる。また、同サーバ装置へ番組のマルチメディア情報を蓄積する場合においても、ホストBからホストAへのセグメントの転送ができなくなる時間は非常にわずかである。したがって、マルチメディア情報入力装置から送信されてきた番組のマルチメディア情報をとりこぼすことはない。

【0238】

〔発明が解決しようとする課題〕で述べた（方式2）では、RDMA領域間でのデータのコピー（移動）やディスクからのデータの再読み出しが必要である。上記のような配信動作や外部からのマルチメディア情報の受信をとぎれなく実行するには、これらの動作を高速に実行する必要があるが、これは非常に困難である。しかし、本発明においては、このような問題も解決される。

10

【0239】

以上に述べたことから、本発明によれば、通信リンク上で障害が発生した場合の迅速な通信リンクの切り替え動作を維持しながら、〔発明が解決しようとする課題〕で述べた従来の自明な通信リンク冗長化方式が持つ問題を解決することが可能である。

【0240】

本発明の装置はコンピュータとプログラムによっても実現でき、プログラムを記録媒体に記録することも、ネットワークを通して提供することも可能である。

【図面の簡単な説明】

【0241】

【図1】背景技術であるPCクラスタシステムの構成例を示す図である。

20

【図2】背景技術であるuDAPLのAPIで用いられる主なオブジェクト群を示す図である。

【図3】uDAPLのRDMA readにより2台の計算機間で通信を行う際の手順を示す図である。

【図4】従来技術の例である（方式1）を説明する図である。

【図5】従来技術の例である（方式2）の概要を説明する図である。

【図6】従来技術の例である（方式2）の動作の詳細を説明する図である。

【図7】従来技術の例である（方式2）の動作の詳細を説明する図である。

【図8】従来技術の例である（方式2）の動作の詳細を説明する図である。

【図9】本発明の課題を解決するための手段の概要を説明する図である。

30

【図10】本発明の課題を解決するための手段の動作の詳細を説明する図である。

【図11】本発明の実施の形態1の装置全体の構成を示す図である。

【図12】本発明の実施の形態1におけるサーバ内データベースが管理する情報を説明する図である。

【図13】本発明の実施の形態1における通信サーバモジュールの構成を示すブロック図である。

【図14】本発明の実施の形態1における蓄積サーバモジュールの構成を示すブロック図である。

【図15】本発明の実施の形態1における通信サーバモジュールの処理を示すフローチャートである。

40

【図16】本発明の実施の形態1における通信サーバモジュールの、端末へのセグメント配信処理を示すフローチャートである。

【図17】図14のステップ11の詳細を示すフローチャートである。

【図18】本発明の実施の形態1における通信サーバモジュール、および、蓄積サーバモジュールの通信経路監視部の処理を示すフローチャートである。

【図19】本発明の実施の形態1における蓄積サーバモジュールの処理を示すフローチャートである。

【図20】本発明の実施の形態1における蓄積サーバモジュールの、蓄積装置からのセグメント読み出し処理を示すフローチャートである。

【図21】図18のステップ77の詳細を示すフローチャートである。

50

【図 2 2】 本発明の実施の形態 2 の装置全体の構成を示す図である。

【図 2 3】 本発明の実施の形態 2 における通信サーバモジュール、および、蓄積サーバモジュールの構成を示すブロック図である。

【図 1】

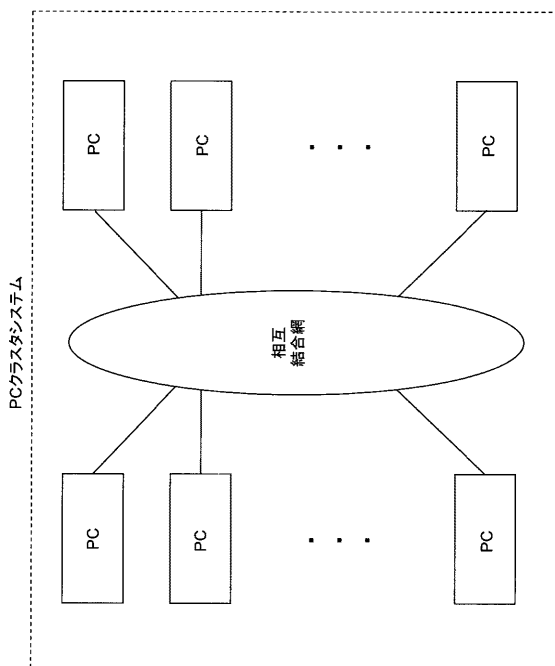


図 1

【図 2】

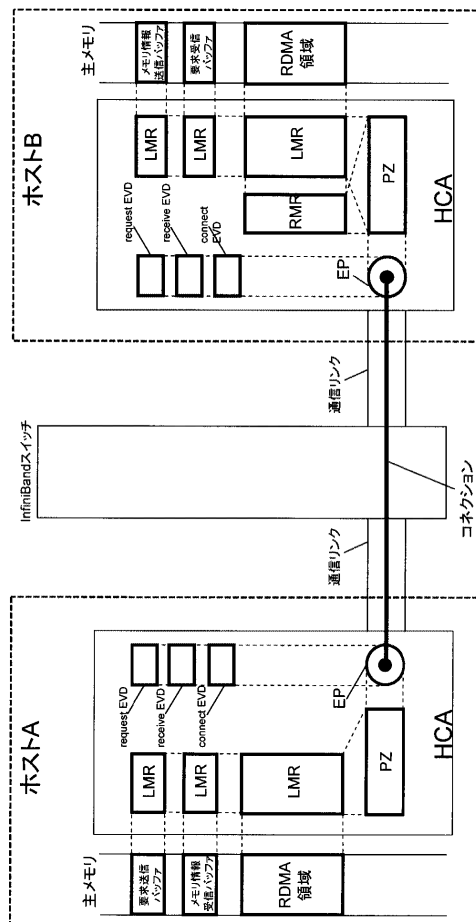


図 2

【図3】

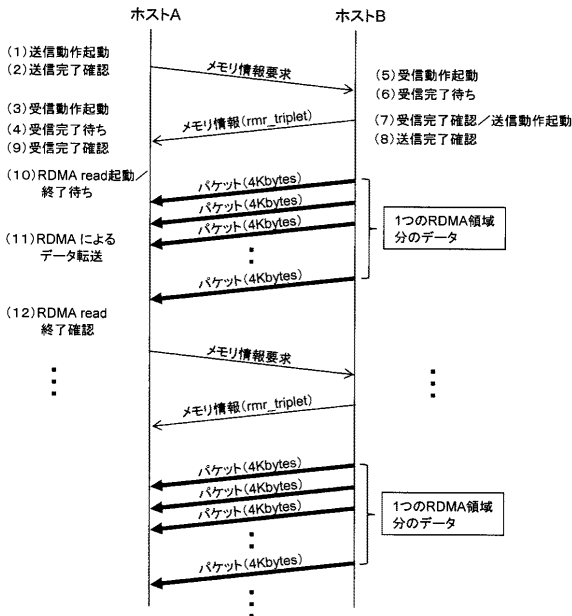


図3

【図4】

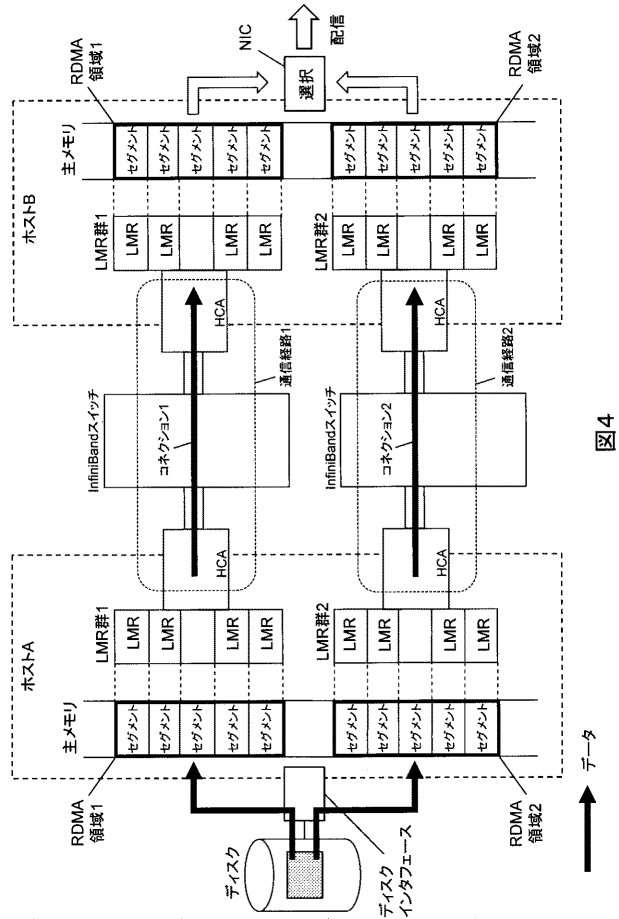


図4

【図5】

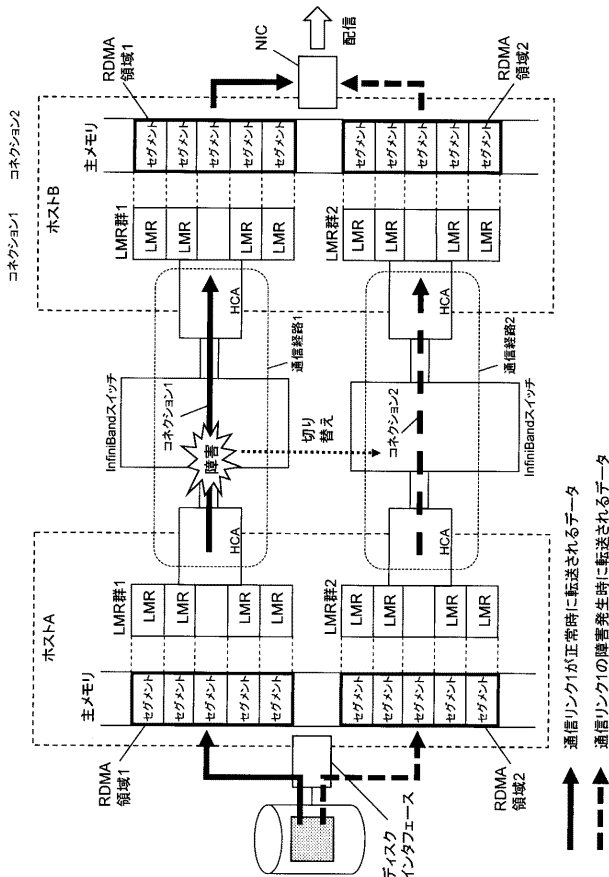


図5

【図6】

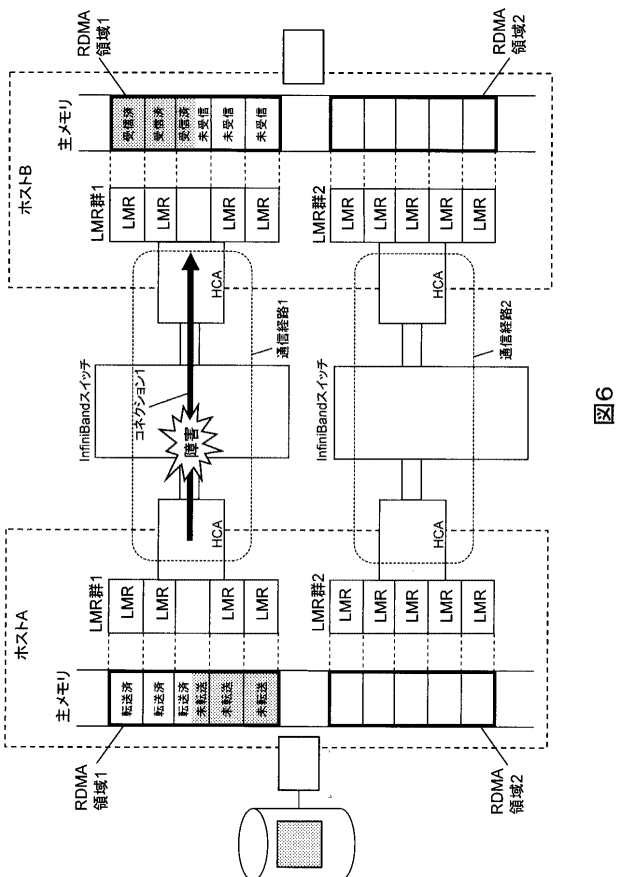


図6

【図 7】

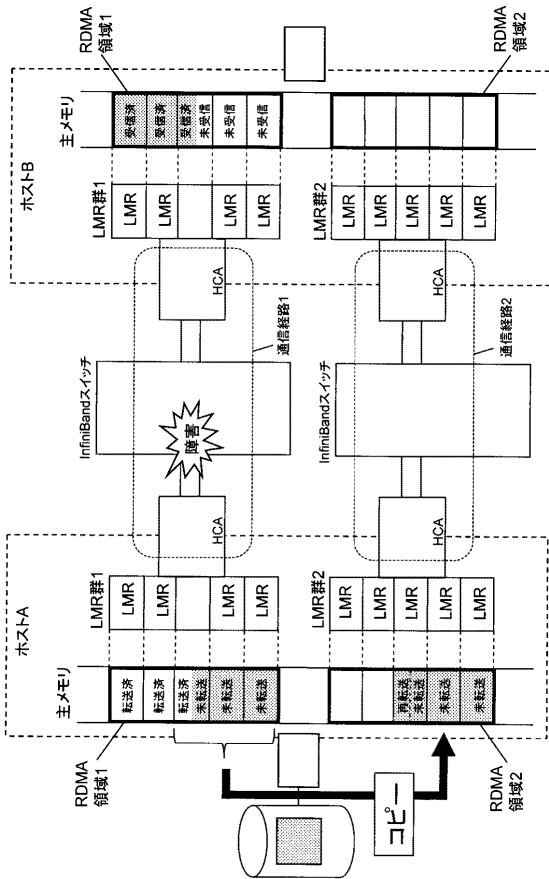


図 7

【図 8】

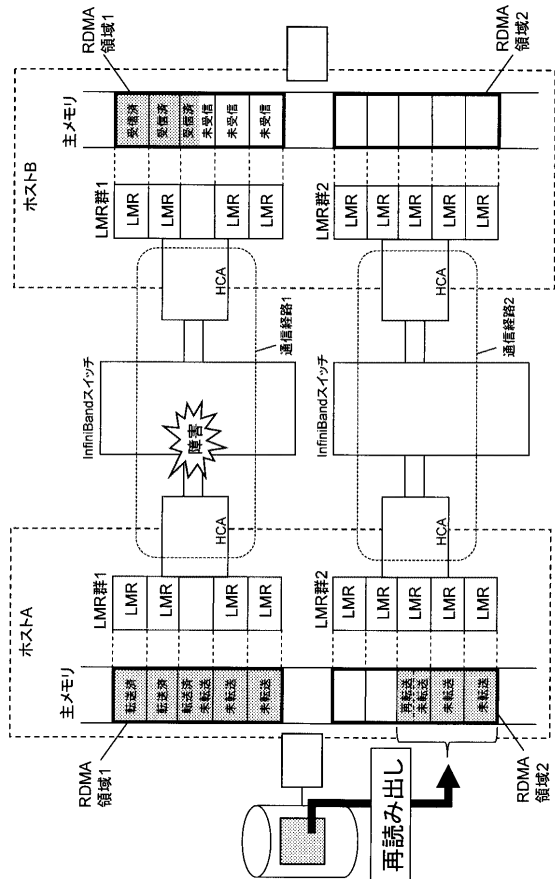


図 8

【図 9】

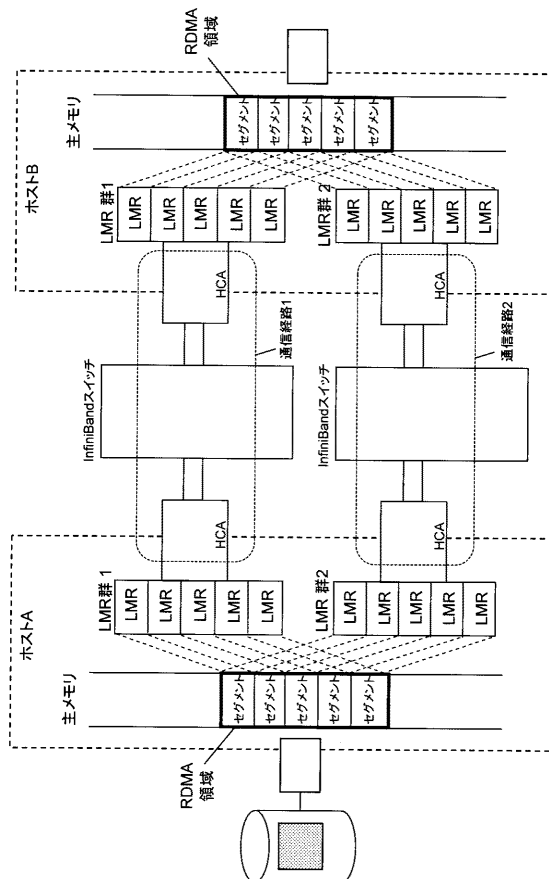


図 9

【図 10】

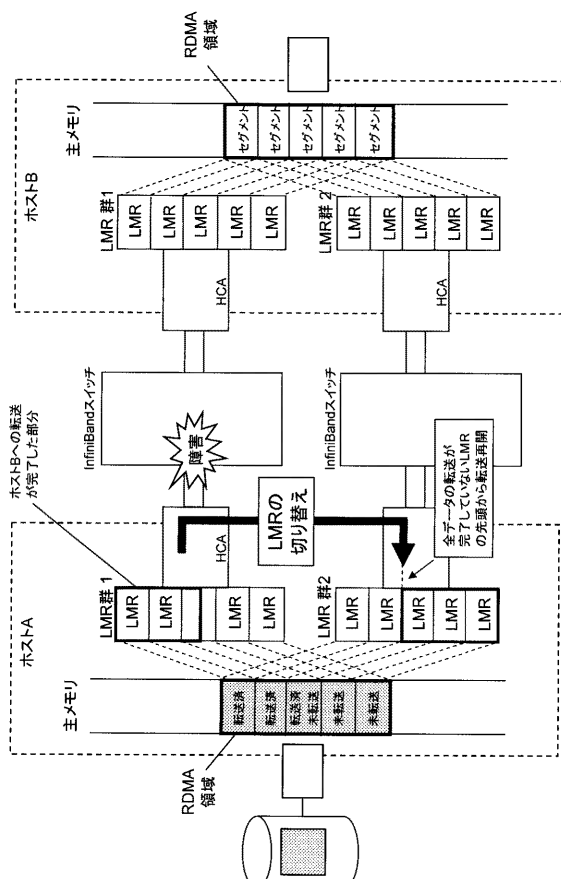


図 10

【図 1 1】

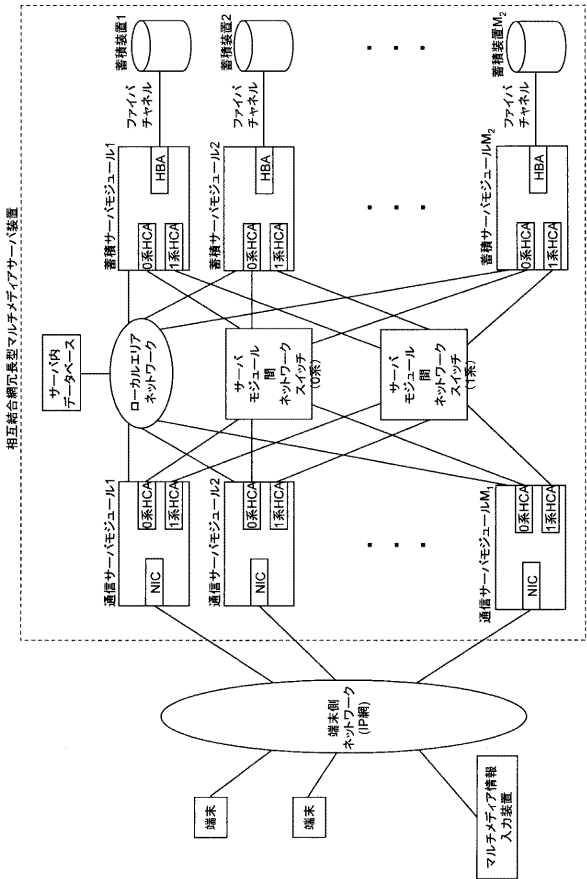


図 11

【図 1 2】

番組格納情報テーブル

番組ID	格納蓄積装置番号
1	5
2	M ₂
⋮	⋮

蓄積装置使用状況テーブル

蓄積装置番号	空き容量 (Kbytes)
1	19719996
2	3877648
⋮	⋮
M ₂	401020512

図 12

【図 1 3】

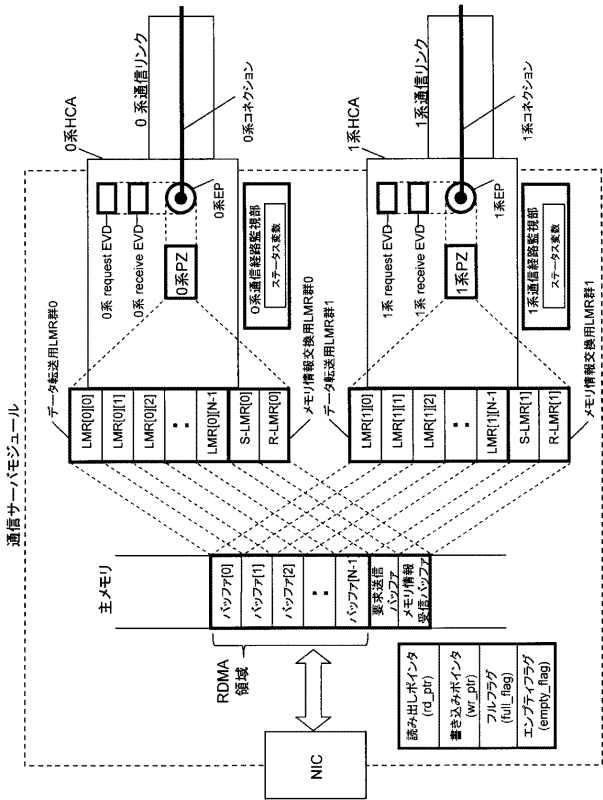


図 13

【図 1 4】

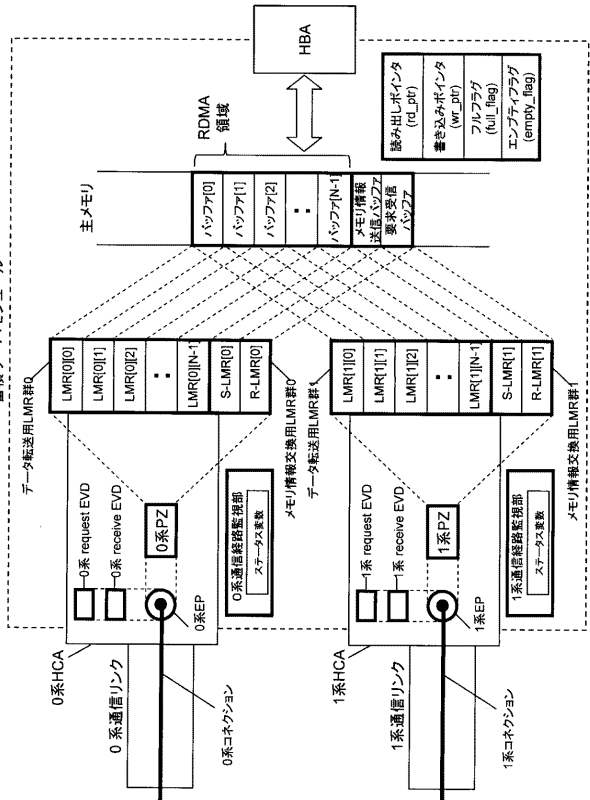
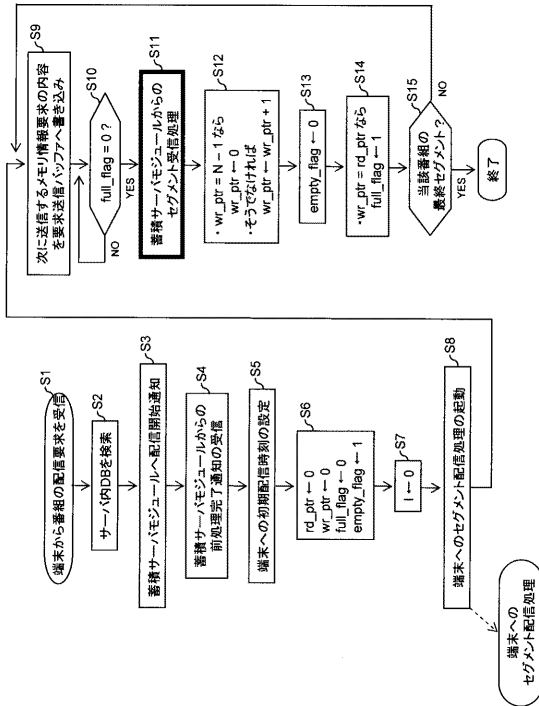


図 14

【図 15】



【 図 1 6 】

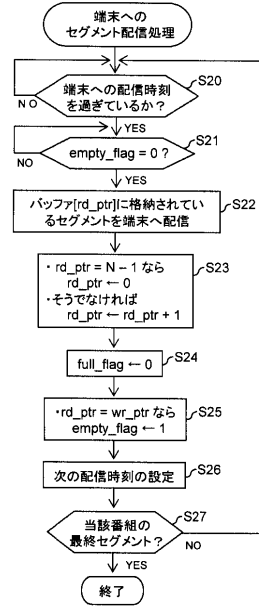
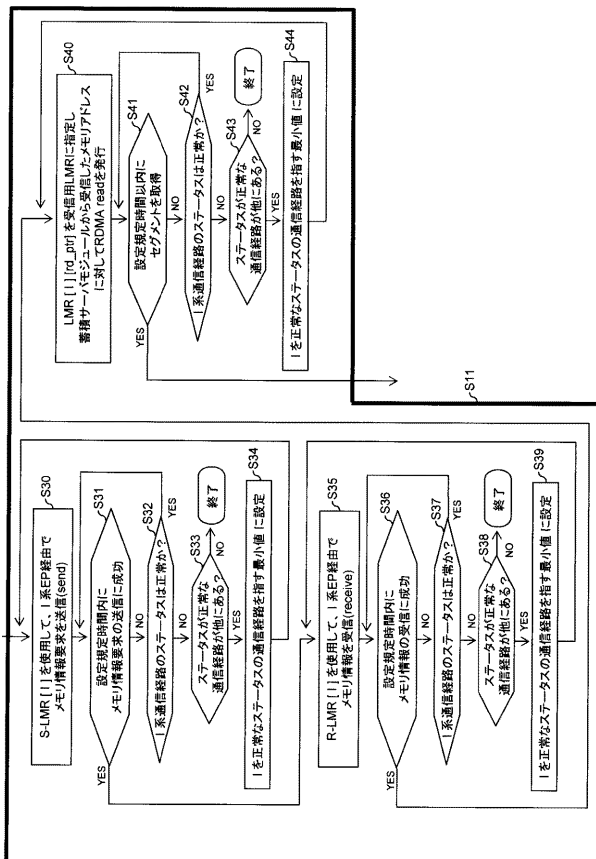
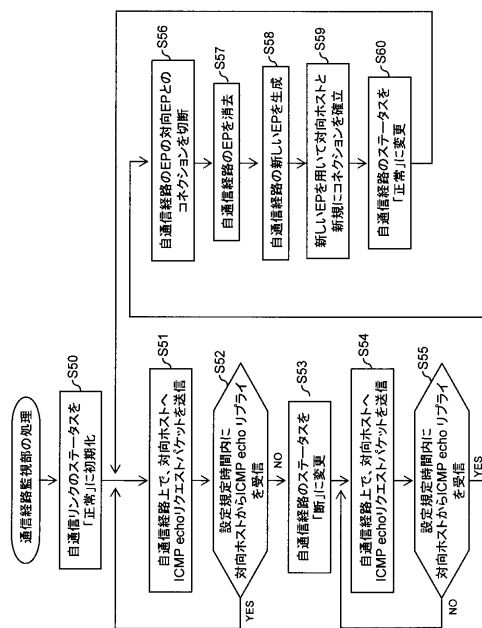


图 16

【図 17】



【 図 1 8 】



81

【図19】

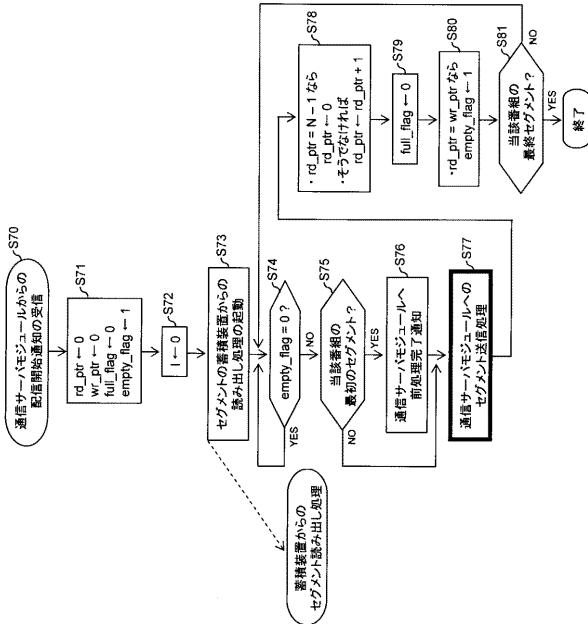


図19

【図20】

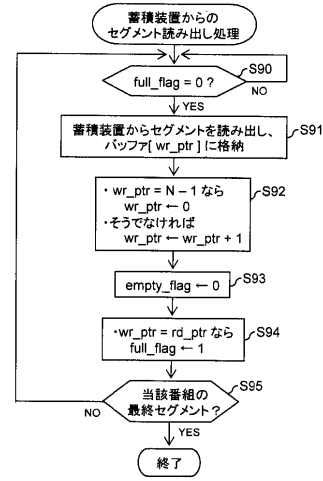


図20

【図21】

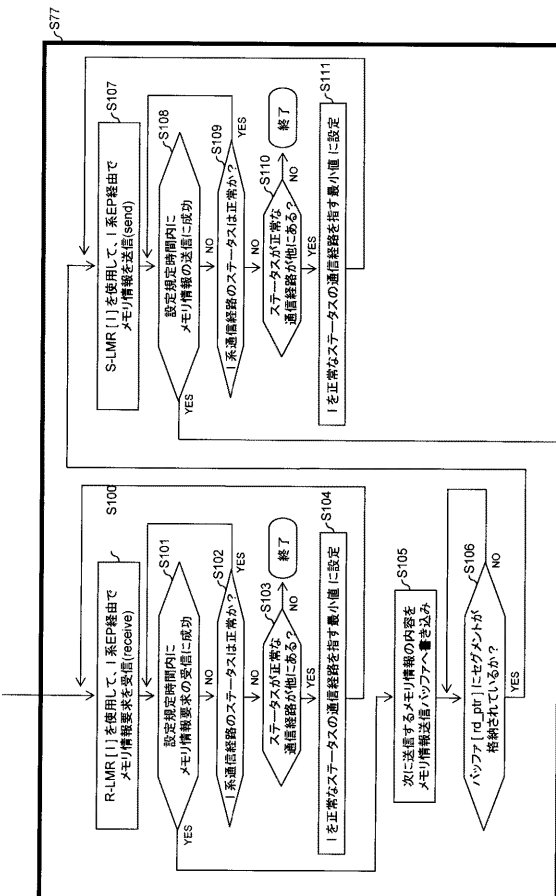


図21

【図22】

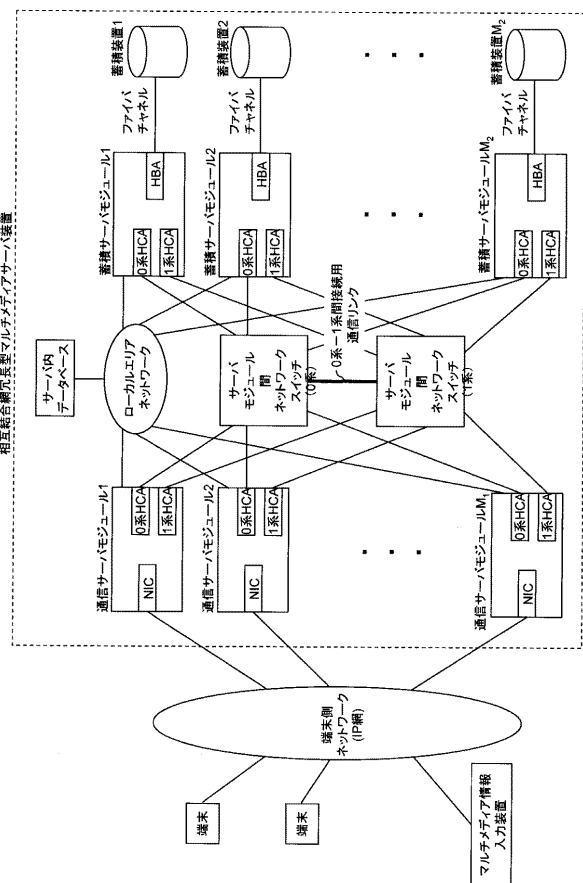


図22

【図 23】

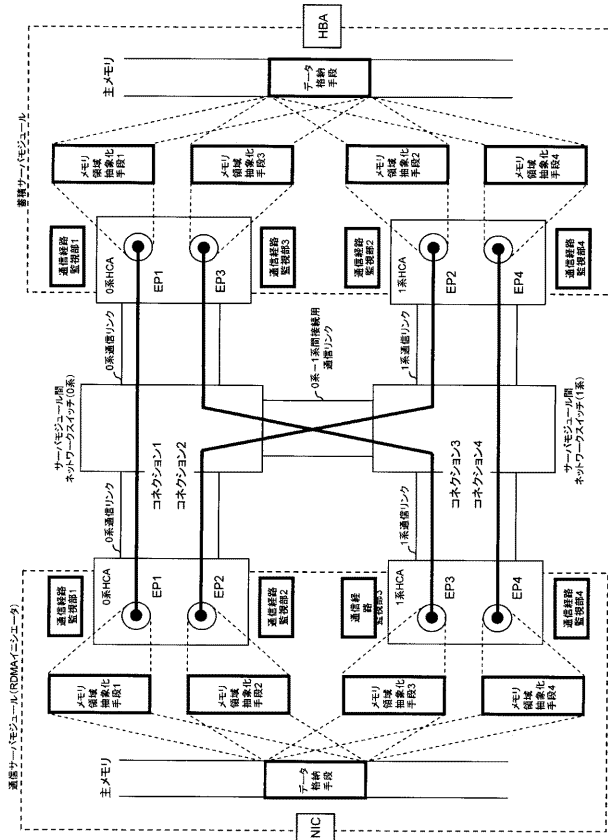


図 23