

グループ通信ミドルウェアの冗長化設計

増田大樹[†] 大谷治之[†] 落合真一[†]
三菱電機(株) 情報技術総合研究所[†]

近年ネットワークが高速・広帯域化したことにより、複数の計算機をネットワークで結合し協調動作を行う分散システムの適用分野が広がっている。分散システムでは複数の計算機に同一データを同報するグループ通信の仕組みが有用である。我々は数十台の計算機を結合した分散システムにおいて、計算機間に多重に設置したLANを用い一部のLAN接続に障害が発生してもシステムの運用を継続できるグループ通信ミドルウェアを設計した。本ミドルウェアはネットワーク上の任意の位置に障害が発生した場合は障害のない経路を用いるように設定を変更し、ノイズや輻射によるパケットの消失があった場合は再送によりデータを確実に届けることができる。本稿ではミドルウェアの設計内容について述べる。

A design of redundant group communication middleware

Hiroki Masuda[†] Haruyuki Ohtani[†] Shinichi Ochiai[†]
Information Technology R&D Center, Mitsubishi Electric Corporation[†]

To construct distributed systems, group communication is useful for reducing the traffic. So many reliable multicast protocols are proposed for wide area network and used in many situations. However, in many of these protocols, the redundancy is supported by routers, and the protocols have no redundancy support in itself. So in local area network if an accident happens on network equipment, there is no way to recover the failure. So we are developing group communication middleware which uses redundant network between computers and has network failover mechanism. In this paper, we describe the design of redundant network and its group communication protocol.

1. はじめに

ネットワークの高速化・広帯域化及び一般化により大規模な科学技術演算から工場の製造ラインまで様々な分野に計算機をネットワークで結合した分散システムの導入が進んでいる。上記のようなシステムでは一対多の通信を効率よくかつ確実にやりたい場合がある。このような時、受信側を一つのグループとしてデータを一括送信するグループ通信の仕組みは通信量を削減する上で有用である。

分散システムの中には高度な耐故障性が求められるものがあり、そのようなシステムでは計算機やネットワークに故障が発生しても処理の継続ができなければならない。

本稿では耐故障性を高めるために計算機間に多重にネットワークを敷設した分散システムに向けて、以下の機能を備えたグループ通信ミドルウェアの設計を行う。

- データを確実に届けるためにパケットの到達確認と再送処理を行う。
- データの送信量がより少なくなるようにする。

- 通信路上の任意の位置に故障が発生しても、各受信計算機との間に正常な通信路が存在すれば、その経路を用いるように経路を切り替える。

2. 背景

2.1. システム概要

ターゲットとしているシステムは数十台の計算機をローカルエリアネットワークで結合した分散システムとなっており、複数の計算機が外部から入力される大量のデータを協調して処理する。

システムは入力されたデータを複数のプロセス・計算機で処理する。一部のアプリケーションは一つのプロセスの処理結果を複数のプロセスが利用してデータの処理を進める。このように複数のプロセス・計算機に同一データを配信するところにグループ通信を用いる(図1)。データは送信時にグループに参入している受信プロセスに確実に配送する。パケットが消失した場合は再送を行う。

送信側のプロセスは処理結果をグループ宛に送信し処理を完了する。受信側のプロセスは各時点の処理内

容に応じて、必要なデータを入手できるグループに参加する。受信側がグループを切り替えて動作するため、データの配送先となる受信計算機の構成は動的に変化する。

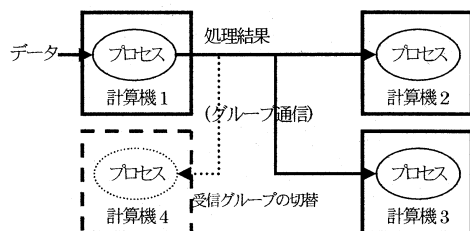


図 1 通信パターン

図 2 は本システムのネットワーク構成である。各計算機はネットワークケーブルの断線に備えネットワークインタフェース(NIC)を複数持っている。またネットワークスイッチ(Switch)の故障に備えシステム中に複数の Switch を配置し、各 NIC はそれぞれ異なる Switch に接続する。さらに、各 Switch 間を接続することで、計算機の任意の NIC 同士で通信できるようにしている。

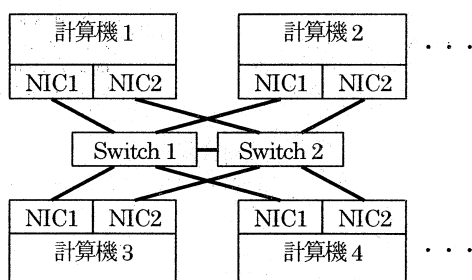


図 2 ネットワーク構成

2.2. 目的

先に述べたネットワーク構成を用い、一部の通信路において断線や switch の故障が発生しても、送信元の計算機から受信計算機までの間に一つでも使用できる通信路があれば通信が成功するグループ通信を実現する。このときデータの送信量を可能な限り少なくする。

2.3. 関連技術

2.3.1. SCTP

SCTP(Stream Control Transmission Protocol)^[4]は本システムのように計算機が複数の NIC を持っているマルチホーミング環境をサポートしたトランスポート層の通信プロトコルである。近年では Linux 2.6 kernel に標準搭載される等、普及が進んでいる。

SCTP は送信元と送信先の間の複数の NIC でそれぞれコネクションを張り、それらをまとめてアソシエ

ションと呼ぶ一つのリンクとして扱う。アソシエーション内部では組み込みのハートビートを用いてコネクションの接続チェックを行い、通信障害を検出したときは他のコネクションを用いるようにフェイルオーバーする。また SCTP は Selective ACK を用いた到達確認と再送制御を行う事で、TCP 同様に通信パケットの到達性保証を行うことができる。

しかし、SCTP は一対一の通信を行う通信プロトコルであるため、グループ通信を行うためには受信計算機数の送信を繰り返す必要がある。

2.3.2. リライアブルマルチキャスト

IP マルチキャスト^[5]を用いると IP ベースで接続された複数の計算機・プロセスに同一データを容易に配信することができる。

IP マルチキャスト自体はパケットが到達保証されないベストエフォート型の通信であるため、IP マルチキャストに信頼性保証機構を加えたりリライアブルマルチキャストプロトコルの研究が進んでいる。システムの規模やアプリケーションに向けて様々なリライアブルマルチキャストプロトコルが作成されており、参考文献[2]では様々なリライアブルマルチキャストプロトコルの分析と比較が行われている。

既存のマルチキャストプロトコルは主にインターネットのように広域分散環境をターゲットにしており、通信経路の故障はルーターの機能で解決している。

2.3.3. NIC の teaming

複数の NIC をサポートした技術に NIC の teaming あるいは channel bonding と呼ばれる技術がある。これはドライバレベルで複数の NIC を一つの NIC に束ねるもので、各 NIC は同一の MAC アドレス、IP アドレスを持って動作する。最近の Linux や Solaris はこの機能を標準で備えており、それぞれ bond モジュール¹や IPMP²でこの機能を実現している。

teaming は NIC の切り替え機能を持っており、使用中の NIC と Switch の間が断線した場合はリンクダウンにより断線を検出し、他の NIC を用いて通信を継続する。

2.3.4. 関連技術の課題

teaming した環境でリライアブルマルチキャストを実施すれば必要な機能の一部は実現できる。しかし、teaming は計算機が接続している LAN ケーブルのリンクアップ・ダウンを用いて経路切り替えを行うため、接続している Switch より遠いところに故障が発生した場合は経路の切り替えを行うことができない。

例えば図 3 の例において×印の箇所が断線した場合

¹ Kernel 2.4.18 以降標準で搭載されている。

² IP マルチバス。Solaris 8 10/00 以降標準で搭載されている。

合、計算機 1 は NIC1 を用いて送信しても計算機 3 にパケットを到達させることはできない。そのため、本システムでは NIC2 に切り替えて送信する必要があるが、計算機 1 の NIC1 はリンクダウンしていないので経路を切り替えることができない。

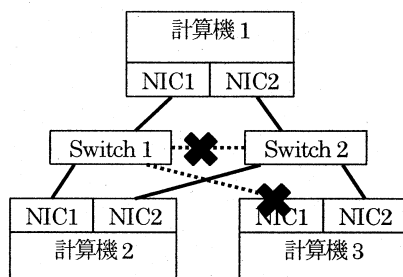


図 3 断線の例 1

2.4. 設計方針

本稿では 2.3.4 で述べた teaming では実行できない経路切り替えを行い、システム要求であるパケットの到達保障と通信量の削減を果たすグループ通信ミドルウェアの設計を行う。

データ通信量削減のために、複数の計算機に同一データを一括送信できる IP マルチキャストを用いる。このとき、データの到達を保証するために、ACK を用いたリライアブルマルチキャストに使用する NIC の切り替え機能を備えたグループ通信ミドルウェアを検討する。

また、LAN のリンクアップ・ダウンではなく、実際にパケットが到達するか否かをチェックすることで、確実に経路の切り替えが行われるようにする。

3. 課題検討

3.1. 送信経路の切り替え

2.3.4 で述べたように、計算機は自身と Switch の間で発生した通信障害だけでなく、Switch と送信先の間の故障が発生した通信障害も検出し、パケットを送出する NIC を切り替えなければならない。また、データ送信量を削減するためには、故障が無いときはパケットをひとつの NIC からマルチキャスト送信する仕組みが必要である。例えば、図 3 の構成で断線故障が発生していない場合計算機 1 は NIC1 からパケットを送信し、図 3 のように断線が発生した場合は NIC2 からパケットを送信するように切り替える必要がある。

また、受信計算機は動的に受信するか否かを切り替えるのでその切替え状況を送信計算機が把握する必要がある。受信計算機の状態を把握しなければ、パケットを受信しない計算機に対して再送制御を行う事にな

る。

3.2. 経路切り替え発生時における同一パケット多重処理の回避

IP 通信では送信に用いた NIC の IP アドレスが送信元計算機として認識される。そのため、経路切り替えを行った場合、受信計算機には送信元 IP アドレスが異なるパケットが到着することになる。

IP マルチキャストでは複数の計算機に同一パケットを送信するため、通信障害が発生しなかった計算機は、経路切り替え前に受信したパケットと経路切り替え後に再送されたパケットを 2 重に受信する事になる。このとき、経路切り替えが発生していることから、この 2 つのパケットは異なる送信元 IP アドレスを持つことになる。

そのため、本グループ通信ミドルウェアは、その 2 つのパケットが同一のものである事を認識し、多重に処理しないようにしなければならない。

4. 設計概要

本章では前記課題を解決するグループ通信の冗長化通信方式について述べる。

4.1. 送信経路の切り替え方式

4.1.1. 通信障害の検出と受信計算機の検出

送信計算機は NIC ごとに受信グループに向けて IP マルチキャストでハートビート通信を行う。受信計算機はハートビートに応答する際に、受信グループに参加して受信を開始するか、受信グループから離脱して受信を停止するか応答する。

このとき、送信計算機は NIC と受信計算機ごとに、ハートビートに応答したか否かをチェックする到達管理表を作成する(表 1)。ハートビートに一定回数以上応答がこなければ、該当する NIC で送信しても通信できないため、通信障害を検出し、該当するエントリに通信不可のマークをする。

ハートビートによる明示的な離脱を行わずに全ての NIC に応答しなくなった計算機は、再度ハートビートに応答するまでエントリを削除する。

表 1 到達管理表

送信に用いた NIC	受信計算機	通信可否
NIC1	計算機 2	×
	計算機 3	○
NIC2	計算機 2	○
	計算機 3	○

4.1.2. 使用する NIC の選択

送信計算機は到達管理表を調べ、接続されている複数の NIC の中から、すべての受信計算機に通信できる 1 枚の NIC を選択し送信する。

通信障害を検出し、到達管理表が更新された時は、別の NIC ですべての受信計算機に通信できるものがあればそれを使用するように送信経路を切り替える。

また一つの NIC で全ての受信計算機にデータを届けることができない場合は複数の NIC を用いて送信するように設定を切り替える。例えば、図 4 の場合は NIC1,2 を両方用いて送信するようにする。

さらに、断線から復旧した場合は通信量を削減するために再び 1 枚の NIC で送信するように設定を変更する。

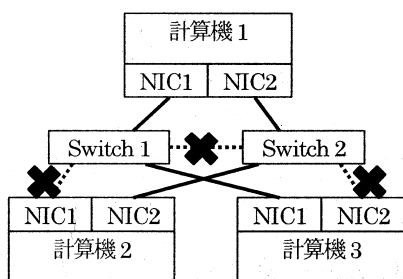


図 4 断線の例 2

4.2. 多重処理の回避

経路切り替えが発生してもパケットの送信元が同一である事を判定するために、パケットに計算機の ID を載せる必要がある。そこで、本通信方式では全ての計算機にそれぞれ IP マルチキャストアドレスを割り振り、ID とする(図 6)。

通信パケットにこの IP マルチキャストアドレスを載せることにより計算機を同定し、多重処理を避ける。

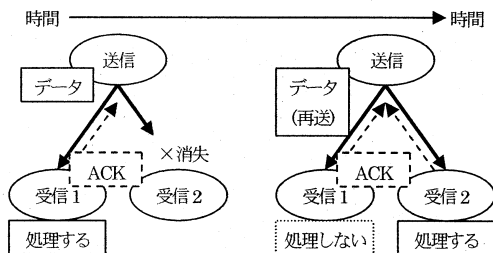


図 5 受信処理

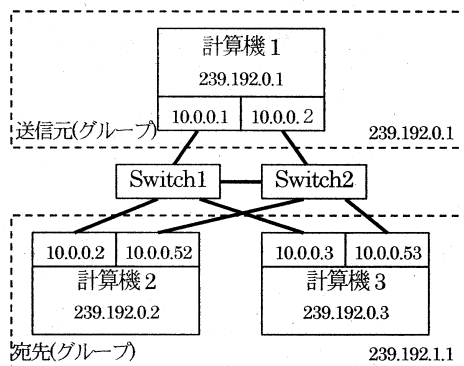


図 6 アドレス構成

4.3. パケット受信の通知

前述のように、本グループ通信ミドルウェアでは計算機ごとに IP マルチキャストアドレスを割り振っている。データパケットを受信した計算機は、パケットに記載されている送信計算機の IP マルチキャストアドレスに向けて、パケットを受信した NIC を用いて ACK を送信する。この ACK には受信計算機の IP マルチキャストアドレスを記載する。

パケットを受信した NIC を用いることで、データパケットが通ってきた通信経路の逆を通り ACK パケットを送信することができる。

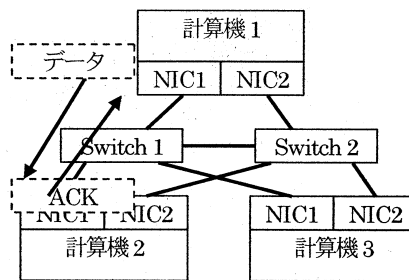


図 7 ACK の応答方法

このとき、送信計算機からマルチキャストされたパケットが受信計算機の複数の NIC に同時に到達する場合があるため、ACK を応答してから一定時間を ACK を送らない時間を定める事で ACK の過剰な送信を防ぐ。

5. 設計詳細

本章では4章で述べた通信方式を実現するための通信プロトコルの設計を行う。

5.1. 概要

本節では4で設計した通信方式を用いて、複数の受信プロセスに確実に同一パケットを配信する通信プロトコルの設計内容について述べる。

同一パケットを受信するプロセスをグループとし、グループにIP マルチキャストアドレスを割り当てる。

データはグループの IP マルチキャストアドレス宛に送信し、パケットの到達を確認する ACK パケットは送信元計算機の IP マルチキャストアドレスに送信する。

5.2. パケットフォーマット

通信パケットに以下の要素を持つヘッダをつける。

- 宛先 (To)
グループ通信の宛先マルチキャストアドレスを入れる。
- データ送信元 (From)
データパケットを送信した計算機のマルチキャストアドレスを入れる。
- パケット送信元(Packet)
パケットを送信した計算機のマルチキャストアドレスを記載する。例えば図6で計算機2がACKを送る場合は239.192.0.2を入れる。
- タイプ (type)
パケットのタイプを入れる。後述するようにタイプにはDataやACK等を定義する。
- アプリID (Id)
送信元アプリケーションの識別子を入れる。
- シーケンス番号 (Seq)
送信元アプリケーションにおける送信順序を示す一連の連番を入れる
- タイムスタンプ (Time)
送信元における送信時間を入力する

宛先 (To)	
データ送信元 (From)	
パケット送信元 (Packet)	
タイプ (Type)	アプリID (Id)
シーケンス番号 (Seq)	
タイムスタンプ (Time)	

図8 通信ヘッダ

どのアプリケーションが送信したデータか判別するために、送信元計算機でヘッダに入力した To, From, Id,

Seq の各フィールドは一連の通信で固定しておく。

5.3. パケット到達確認と再送

到達確認と再送は以下の手順で行う。(図9)

1. [送信側] タイプにDataを入れて送信する
2. [受信側] タイプにACKを入れ、Dataパケットのパケット送信元に応答パケットを送信する。
3. [送信側] 受信したACKパケットの送信元にパケットが到着した事を記録する。
4. [送信側] 一定時間経過してもACKが届いていない計算機があればDataパケットを再送する。
5. [送信側] 一定回数再送を繰り返してもACKが届かない計算機があれば再送を打ち切り、経路管理表からその計算機を取り除く。

再送を行う時に使用するNICを切り替え、経路管理表のアップデートを行う。

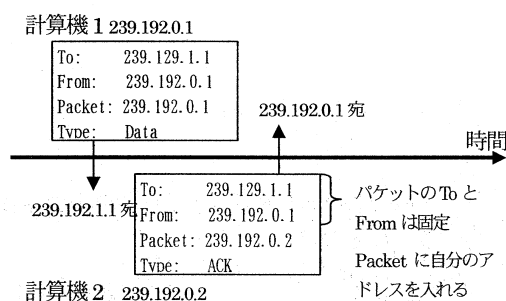


図9 到達確認・通信の流れ

5.4. ハートビート通信および初期化处理

ハートビート通信として、送信計算機は周期的にパケットをマルチキャスト送信する。受信計算機が起動した時は、このハートビートパケットに応答する事により、再送制御に必要なシーケンス番号の初期化处理を開始する。以下にハートビート通信の送信から初期化までの流れを述べる。(括弧内はパケットのタイプ)

1. [送信側] 周期的に受信グループのマルチキャストアドレスに状況を問い合わせるハートビートパケットを送信する。(query)
2. [受信側] queryを受信したら該当するNICが使用可能であることを通知するパケットを送信元に返信する。(usable)
3. [送信側] 初期化を行っていない受信計算機からusableが届いたら最新のシーケンス番号をusableの送信元に返信する。(init)
4. [受信側] initを受信したことを送信側に応答する。(initack)

このとき、queryは受信グループ全体のIPマルチキャストアドレス宛に送信し、その他の通信は計算機個別のIPマルチキャストアドレスに送信する。

何らかの原因で init/initack パケットが消失した場合、送信側は initack が返らないことから init の再送を行う。query/usable パケットが消失した場合は送信側が受信側の起動を気づかないため、query の送信から繰り返す。

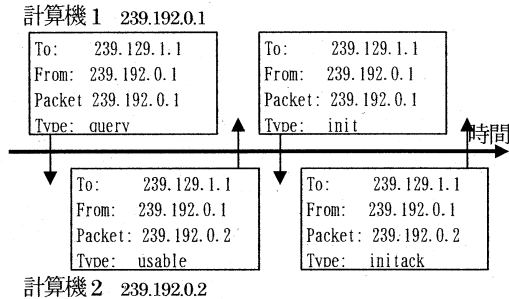


図 10 初期化処理

6. 実装概要

現在これまで示してきた設計の内容を元にグループ通信ミドルウェアを試作している。本章では実装の概要について述べる。

6.1. 受信デモン

本システムでは同一パケットを受信する複数種類の受信アプリケーションが同じ計算機上で稼動する可能性がある。また、受信アプリケーションの負荷が高くなった時にハートビートに応答できなくなると不要なフェイルオーバーを引き起こすことになる。

そこで、プロトタイプではネットワークからの受信とパケットのバッファリングを行う受信デモン設置し、受信アプリケーションとプロセス間通信を行うようにした。

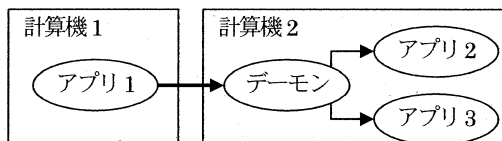


図 11 受信デモン

この構造を用いることで、送信側プロセスは受信側プロセス一つ一つに対して再送制御する必要がなくなり、受信アプリケーションの負荷が高くなっても確実にハートビートに応答することができる。

6.2. ACK の受信

一つの計算機中に複数の送信アプリケーションが存

在することから、プロトタイプでは前記受信デモンにて ACK を受信し、ACK 受信完了をプロセス間通信で送信アプリケーションに通知することにした。データ送信から ACK 受信までの流れは図 12 のようになる。

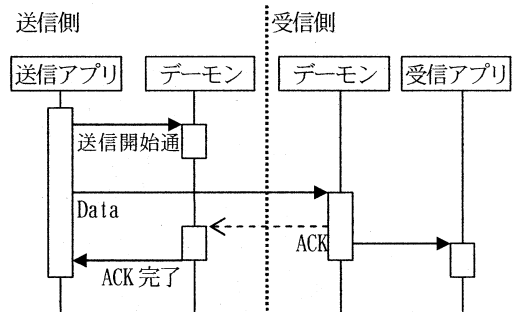


図 12 通信の流れ

7. おわりに

本稿では複数の計算機をネットワークで結合した分散システム向けに複数のアプリケーションに同一データを配信するグループ通信ミドルウェアの設計について述べた。以下に本ミドルウェアの特徴を記す。

- 通信経路上の任意の箇所に障害が発生しても、使用する NIC を適切に切り替える事ができる。
- 各計算機に IP マルチキャストアドレスを割り当て、前記アドレスを識別子として用いる。
- 前記マルチキャストアドレスを応答の送信先に用いる。

今後 6 章で示した試作を実装し評価を行っていく予定である。

参考文献

- [1] 下國治 等：インターネットでの並列分散処理の方式検討, SWoPP '99 CPSY99-67 (1999)
- [2] 木下真吾：リライアブルマルチキャスト技術の最新動向, 電子情報通信学会論文誌 B Vol.J85-B No.11 pp.1819-1842 (2002)
- [3] Dave Kosiur 著, 菊田幸雄 監訳: マスタリング TCP/IP IP マルチキャスト編, オーム社(1999)
- [4] R.Stewart et al., "Stream Control Transmission Protocol", IETF RFC 2960, (2000)