

# 3G-9 OZ+ : オブジェクト指向開放型分散システム<sup>1</sup>

— OZ+ における Multicast RPC —

塚本 亨治 (電総研)<sup>2</sup> 梶浦 広行 (シャープ)<sup>3</sup> 水谷 功 (住友電工)<sup>4</sup> 篠原 弘樹 (松下電器)<sup>5</sup>

## 1 はじめに

OZでは、通信プロトコルとしてOSIコネクションレスネットワークの上に、仮想計算機間で信頼性のあるデータ通信を確保するためのプロトコル、オブジェクトの所在情報管理を行なうプロトコル、及びオブジェクトを呼び出すプロトコルを実現していた[1]。しかし、アプリケーションが通信プロトコルに要求するものは、オブジェクト間のメッセージの送受信による遠隔手続き呼び出し(RPC)のためのプロトコルである。そこで、オブジェクト指向型分散システムに適したRPCの設計を行なった。

本稿では、OZ+におけるRPCとそれを拡張したMulticast RPCについて述べた後、Multicast RPCを用いた例として、オブジェクトの管理を行なうアプリケーションについて述べる。

## 2 OZ+におけるMulticast RPC

### 2.1 OZ+におけるRPCプロトコル

OZ+では、あるオブジェクトから別のあるオブジェクトへメッセージを送信することが、オブジェクトからオブジェクトへ処理を依頼(手続き呼び出し)することになり、実行を依頼するオブジェクトと実際に実行するオブジェクトが別の仮想計算機(VM)に存在する場合には遠隔手続き呼び出し(RPC)となる。

RPCのプロトコルシーケンスを以下に示す(図1)。但し、処理を依頼するオブジェクトをUID1、処理を実行するオブジェクトをUID2としている。

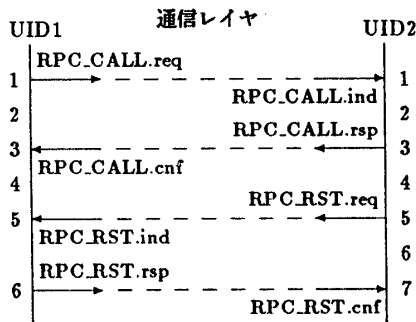


図1: RPCプロトコル

#### ・手続きの呼び出し側 (UID1)

1. UID2にメッセージをRPC\_CALL.reqとして送信する
2. RPC\_CALL.cnfが返ってくるまでブロックされる
3. UID2よりRPC\_CALL.cnfを受信する
4. 別の処理を続ける(ことができる)
5. UID2より実行の結果をRPC\_RST.indとして受信する
6. UID2にRPC\_RST.rspを送信し、RPCは完了する

#### ・手続きを実行する側 (UID2)

1. UID1よりRPC\_CALL.indとしてメッセージを受信する
2. 受け取ったメッセージが自分のメソッドで実行できるかチェックする
3. チェックした結果をRPC\_CALL.rspとしてUID1に送信する
4. 受け取ったメッセージに従って手続きを実行する
5. UID1に実行の結果をRPC\_RST.reqとして送信する
6. RPC\_RST.cnfが返ってくるまでブロックされる
7. UID1よりRPC\_RST.cnfを受信し、RPCは完了する

ここで、処理の依頼のプロトコルと結果を返すプロトコルを分離し、それぞれ確認型プロトコルとしたのは以下の理由による。

1. 処理の依頼と結果を返すプロトコルを別にしておかないとデッドロックが起こる可能性がある
2. メッセージを受け取ったオブジェクトは自分のメソッドで実行できるかチェックする必要がある

さて、RPCで結果を返す時に、どの相手に結果を返せばよいのか一意に決まらない時がある。例えば、VM内に存在するオブジェクトがタイプサーバーからロードされたもののような場合、タイプサーバーに登録されているUIDと同じUIDがVM内でも割り振られる。このようなオブジェクトがRPC要求をすると、手続きを実行する側は結果を返すときに別のオブジェクトに結果を返してしまうことがある(図2)。

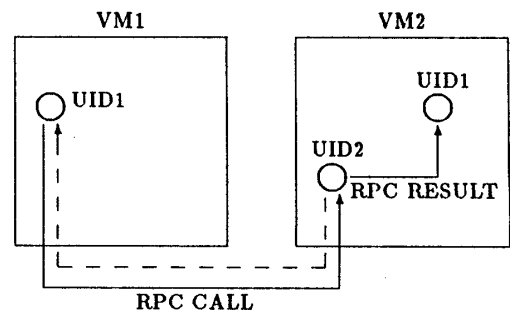


図2: RPCを実現する上での問題

このような問題を避けるために、RPCの送信元のオブジェクトのUIDは以下の制限をつけることにする。

- ・送信元のUIDはシステム内で一意に定まるものでなければならない

この制限によって、RPCの結果を返す相手を一意に決定することができる。

<sup>1</sup>OZ+ : Object Oriented Open Distributed System — Multicast RPC for OZ+ —

<sup>2</sup>Michiharu TUKAMOTO (Electrotechnical Laboratory)

<sup>3</sup>Hiroyuki KAZIURA (SHARP Corporation)

<sup>4</sup>Isao MIZUTANI (SUMITOMO Electric Industries, Ltd.)

<sup>5</sup>Hiroki SHINOHARA (MATSUSHITA Electric Industrial Co., Ltd.)

## 2.2 Multicast RPC

RPCの宛先を1つに限定せず、複数に拡張されたものをMulticast RPCと呼ぶ。しかし、RPCのユーザ側ではMulticastとUnicastとを区別せずに同じ方法で扱いたい。そこで、宛先のオブジェクトをグループ化して、一つの仮想的なオブジェクトに対応させ、要求元のオブジェクトはグループ化されたオブジェクト(そのUIDをGIDと呼ぶ)にメッセージを送る。GIDはメッセージを受けると、RPC\_CALL.reqを下位の通信レイヤ(RPC, RBT[4])に送り、通信レイヤで複数の宛先に分割されて、各送信先にRPC\_CALL.indが届けられる。各RPC実行側からのRPC\_CALL.rspは、通信レイヤでまとめられ、要求元には一つのRPC\_CALL.cnfが届く。実行の結果は手続きの実行側が各々個別に要求元に返す。(図3)。

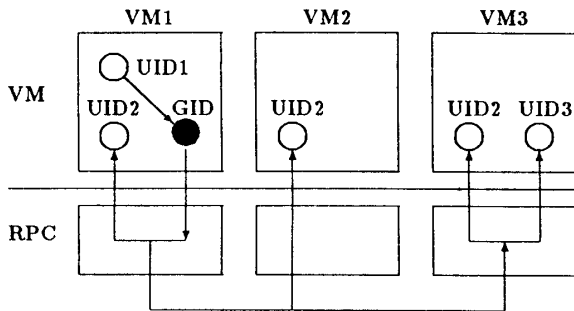


図3: Multicast RPC

RPCの通信レイヤでは、オブジェクトからの要求をグループ化された各々のオブジェクトに配送するため、どのVMのどのオブジェクトがどのグループに属しており、IDとしてどのGIDが割り振られているのか、知っている必要がある。そのため、各RPCではGIDとUIDおよびVMIDとのマッピング表を持っている。

マッピング表の管理は各VMを統括するVM(Mgr)[3]が行なう。そのため、マッピング表はMgrにマスターがあり、グループの追加・削除等、表に変更が生じた場合には、変更が生じたVMがMgrにその旨を知らせる。VMから変更通知があった場合、Mgrはマッピング表を修正したのち、各RPCに表の更新を指示する。

## 3 Multicast RPCによるオブジェクトの移動管理

OZでは、計算の過程において、オブジェクトがVMの間を次々に移動するため、どのオブジェクトがどのVMに存在しているか動的に管理する必要がある。このため、これまでではヒントイングという方法でその管理を行ってきたが[2]、オブジェクト間でのデータ転送の効率などに問題があった。そこで、Multicast RPCを活用してオブジェクトを動的に管理するシステムを設計した。以下に、その概略を示す(図4)。

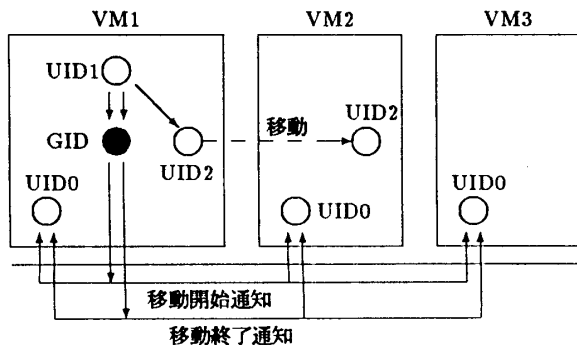


図4: Multicast RPCによるオブジェクトの移動管理

1. UID1はGIDを通して各VMを管理するオブジェクト(UID0)にUID2への送信をロックする旨を通知(移動開始通知)する
2. 各UID0はVM1のUID2が移動しても問題がないかチェックを行ない、問題がなければUID2への送信をロックし、その結果をUID1に返す
3. UID1が結果を受信する前に、UID2にメッセージの受信があった時は、そのメッセージを一時保留しておき、受信確認を相手に通知する
4. UID1は1つでも問題ありとの結果が返ってきたら、各UID0にUID2への送信のロック解除をする旨を通知し、しばらくして1に戻る
5. UID1は全てのUID0から問題なしとの結果が返ってきたら、UID2をVM2に移動させる
6. UID2はVM2に移動する
7. UID1は各UID0にUID2への送信のロックを解除する旨を通知(移動完了通知)する
8. UID2は一時保留してあるメッセージがあればそれを実行する

この方法によるオブジェクトの動的な管理には、ヒントイングによる方法に比べ次の特長がある。

1. ヒントイングをしなくて済むので、データ転送の無駄が省ける
2. 各VMが持っているオブジェクト管理テーブルを一度に書き替えるので、分散ガーベジコレクションが行ないやすい

## 4 おわりに

OZ+におけるRPCとそれを拡張したMulticast RPCの概要と、Multicast RPCを利用してオブジェクトを動的に管理する方法を示した。RPC及びMulticast RPCについては一部実装を行ない、プロトタイプが動作中である。今後は、Multicast RPCとアプリケーションのインターフェイスを検討し、ここで示したオブジェクトを動的に管理するシステムなど、Multicast RPCを用いたアプリケーションを開発していく予定である。

なお、本研究は通産省大型プロジェクト「電子計算機相互運用データベースシステム」の一環として行なわれている。

## 参考文献

- [1] M. TSUKAMOTO et al., The Architecture of Object-Oriented Open Distributed System: OZ, Interoperable Information Systems ISIIS '88, Ohmsha, PP153-166 (1988,11)
- [2] 塚本他, OZ: オブジェクト指向開放型分散システムアーキテクチャ - OZ+分散カーネルにおけるオブジェクトの所在情報管理 -, 情報処理学会 第41回全国大会, 6Q-1 (1990,9)
- [3] 塚本他, OZ+: オブジェクト指向開放型分散システム - オブジェクトの分散管理 -, 情報処理学会 1990年代の分散処理シンポジウム論文集, PP57-64 (1990,11)
- [4] 塚本他, OZ+: オブジェクト指向開放型分散システム - OZ+通信系の拡張 -, 情報処理学会 第42回全国大会, (1991,3)