

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

6Q-1

--- OZ+分散カーネルにおけるオブジェクトの所在情報管理 ---

塚本享治(電総研) 近藤貴士(シャープ)
水谷功(住友電工) 篠原弘樹(松下電器) 梶浦広行(シャープ)

1 はじめに

現在、通信規約の標準化により異機種計算機間の相互接続が可能となってきた。しかし、拡張性などの点で現在の標準化の方法は十分とはいえず、より高度な分散処理アーキテクチャが望まれる。われわれは、処理対象をすべてオブジェクトとして表現する分散処理システム OZ によって拡張性に富む分散処理が可能となることを示してきた。OZ では、オブジェクトがネットワーク上を自由に移動しうするため、動的なオブジェクト所在情報管理機構が必要となる。本稿では、OZ に引続き開発中の OZ+ で行っているオブジェクトの所在情報管理機構について述べる。

2 分散カーネルの機能

OZ では、手続き、文字列、プロセスなどをすべてオブジェクトとして捉え、これらを同様のデータ構造で表現する。ネットワーク上にはオブジェクトの相互作用を処理する仮想計算機が多数存在し、オブジェクトはこれらの間を移動したりこれらをまたがった参照を行う。オブジェクトには仮想計算機の内部からのみ参照されるローカルオブジェクトと、他の仮想計算機からも参照され得るグローバルオブジェクトがある。グローバルオブジェクトには ID が付与されている。ある仮想計算機において他の仮想計算機へのオブジェクトの移動あるいはメッセージの伝達を検出すると、分散カーネルなる機能部に対し宛先オブジェクト ID を指定した送信要求を行う。ここでは、宛先オブジェクトの位置は意識されない。分散カーネルでは、宛先オブジェクト ID からそのオブジェクトが存在する仮想計算機の ID を求め、それに対してオブジェクトあるいはメッセージの送信を行う。OZ では、オブジェクトが動き回ることを許しており、また宛先となりうる対象の数が非常に多いため、従来のディレクトリや、ブロードキャストなどの方法を送信要求発生毎に用いることはできない。そこで、OZ+ 分散カーネルでは既に他の仮想計算機に移動済みのオブジェクトを宛先とした送信要求に対処するためヒンティング機構を設け、さらに各送信要求発生毎に送信内容を捕捉し以後必要になるであろうオブジェクト所在情報を保存・管理する。オブジェクトの所在情報管理に用いる表をオブジェクトアドレス表と呼んでいる。

3 ヒンティング処理

ヒンティングは移動可能な宛先オブジェクトに対して高信頼にデータ転送を行う機構である(図1)。

OZ においてヒンティングはフォワーディングに比べ次の利点がある。

信頼性の高い通信が可能 最終的には実際に宛先オブジェクトが存在する仮想計算機に対し直接データ送信を行うためである

高速の通信が可能 毎回の送信にいちいち経由処理をせずにすむ

転送に関わる情報量が少なく済む 自仮想計算機が利用していないオブジェクトについての所在情報を保存せずにすむ

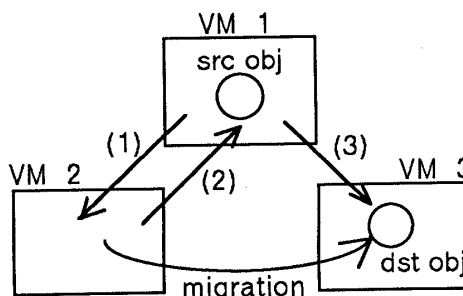


図1. オブジェクトヒンティング

4 宛先オブジェクト

OZ 分散カーネルでは、送信要求時宛先として指定されるオブジェクトは常に他仮想計算機にあるグローバルオブジェクトであり、ID によって指定される。宛先 ID は、それを送信元が知っているという事実から、次のいずれかである(図2)。

1. 自仮想計算機内のオブジェクトを宛先とした送信を行った送信元オブジェクトの ID
2. 自仮想計算機から他仮想計算機へ移動させたオブジェクトの ID
3. 1の送信によって送られてきたオブジェクトあるいはメッセージが参照しているオブジェクトの ID
4. ディレクトリによって名称などからのマッピングによって得られたオブジェクトの ID

1,2,3については、送信要求の内容を捕捉することにより常に宛先オブジェクトの所在を知ることができ

る。4については、ディレクトリにオブジェクトの所在情報を含ませておき、送信処理前にその所在情報を分散カーネルに予め通知しておくことが適当であると考えられる。分散カーネル自身がディレクトリを利用することはない。

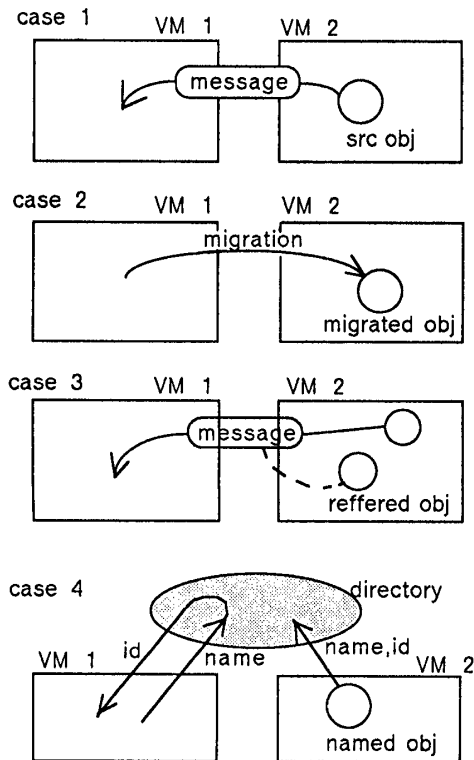


図2. 宛先となり得るオブジェクト

5 オブジェクト転送シーケンス

前述のヒンティング処理を含め、オブジェクト所在情報を管理したオブジェクトあるいはメッセージ転送処理は次のようになる。

[送信側]

1. 分散カーネル利用者が分散カーネルに対しデータ送信要求を行う。
2. オブジェクトアドレス表（以後単に表という）から宛先オブジェクトが存在する仮想計算機のIDを求める。
3. 送信するオブジェクト（あるいはメッセージ）が参照しているすべてのオブジェクトの所在情報を表から求め、PCIにセットする。
4. 求めた宛先仮想計算機へPDUの送信を行う。送信側からのPDUには送信元オブジェクトID、宛先オブジェクトID、転送するオブジェクトID、参照されるオブジェクトのIDが含まれる。
5. 宛先からの応答によって、次の処理を行う。

- 送信成功ならば、宛先オブジェクトが応答を返した仮想計算機に存在すると表に登録する。さらに送信要求がオブジェクト転送のときは、転送されたオブジェクトも同様に表に登録する。その後、分散カーネル利用者に送信成功を通知する。
- 送信失敗ならば、（宛先オブジェクトに対するエントリを表から削除し、）送信失敗を利用者に通知する。
- ヒンティング通知を受けたときは、宛先オブジェクトの移動先仮想計算機に再度PDUを送信する。

[受信側]

1. 送信元仮想計算機からPDUを受信する。
2. 宛先オブジェクトが自仮想計算機内にあるか表を検索し、ない場合は次の処理を行う。
 - 他の仮想計算機に移動済みのときは、ヒンティング通知を送信元仮想計算機へ送り、宛先オブジェクトの移動先仮想計算機のIDを通知する。
 - 宛先オブジェクトの所在情報を持たないときは、送信失敗を送信元仮想計算機へ通知する。
3. オブジェクト転送の場合は、送られてきたオブジェクトを表に登録する。
4. PCIを走査し、参照されているオブジェクトを表に登録する。
5. 送信元へ送信成功を通知する。
6. 分散カーネル利用者へ受信通知を行う。

6 まとめ

OZ+分散カーネルで用いている動的なオブジェクト所在情報管理方法について説明した。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-オブジェクト指向型分散プログラミング言語とその実装-」, 情報処理学会プログラミング言語研究会, 21-4(1989.6)

[3] 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-LLCタイプ3を活用する通信アーキテクチャ、実装、およびその評価-」, 情報処理学会マルチメディア通信と分散処理研究会, 43-4(1989.9)

[4] 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-オブジェクト指向型分散プログラミング言語の複数ユーザ環境への拡張-」, 情報処理学会プログラミング言語研究会, 22-4(1989.10)