OZ+のオブジェクト通信機構の評価

1P−4

濱崎 陽一 、塚本 享治
(電子技術総合研究所)

1) はじめに

オブジェクト指向開放型分散システムＯＺ+は、異機種の計算機がネットワークで接続された開放型の分散環境のもとで、マルチユーザを対象としたオブジェクト指向システムで、昨年度に完成し[1,2]、現在その性能評価を進めているところである。ここでは、オブジェクトの通信機構の性能測定を行った結果とその評価について述べる。

2) ＯＺ+の通信機構とその実装

ＯＺ+は、複数の計算機(ステーション)がネットワークで接続された環境で動作し、各ステーションのうえに実現された仮想マシン(ＶＭ:Virtual Machine)の集合体が、ユーザにはあたかも一つの計算機上に有るかのように見える。ＯＺ+のオブジェクトはSmalltalk[2]のオブジェクトに似ていて、ＯＺ+システムの中の数値、文字列、プログラム、メッセージ、スタックなど全てのものがオブジェクトとなっている。

図1にＯＺ+の階層モデルによるソフトウェア構成を示す。下位の層から順に説明する。

ＲＢＴは、複数のブロックに分割しなければならないような大きなサイズのデータについても高信頼性が得られるように通達確認や順番が入れ替わった分割データの並べ替えなどを行う。プロトコルにはBLASTプロトコルを用いており、転送に失敗した部分のみの再送ですむようになっている。通信の宛先となるＶＭ名とネットワーク上の宛先であるステーションＩＤとの対応表はここで管理される。

ＲＰＣは、リモート手続き呼び出しを実現する。呼び出しに対して結果が帰ってくるだけの通常のＲＰＣと異なり、呼び出しが正常に行われたかどうかの確認(Confirmation)が結果に先だって呼び出し側に返される。プロトコルの動作については、後に述べる。ＲＢＴとＲＰＣは通信モジュール(ＣＭ)を構成し、1つのＵＮＩＸプロセスとして実装されている。ＣＭは、ステーションにそれぞれ1つづつある。

ＯＺ+のプログラムは、ＯＺ言語で記述され、コンパイラによってバイトコード(中間コード)に変換される。ＯＺ+では計算機の機種毎にバイトコードを解釈、実行するプログラムであるバイトコードインタプリタ(ＢＣＩ)を開発することにより、異機種間の相互運用性を実現している。ＢＣＩが扱う全ての実体はオブジェクトであるが、ＢＣＩで実行されているＯＺ+のプログラム間の通信を行うには、複雑な構造を持つオブジェクトをその構造を保ったままで、1次元のデータに変換/復元する必要が有る。それを行うのがＳＴＵＢである。ＢＣＩとＳＴＵＢはＶＭを構成し、ＵＮＩＸプロセスとして実装されている。1つのステーションに複数のＶＭを実現することが可能で、ＶＭはＣＭとネットワークを介して相互に通信できる。

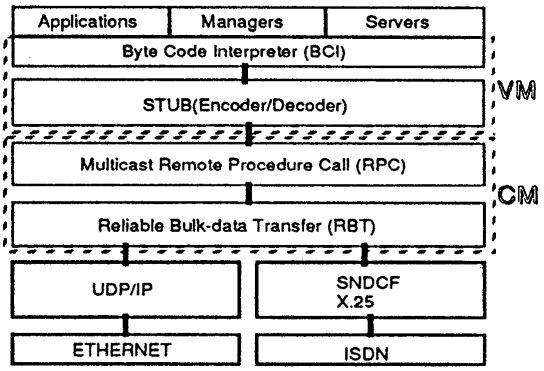

図1 ＯＺ+のソフトウェア構成

図2は、オブジェクト間のメッセージパッシング(オブジェクト呼び出し)の正常時のプロトコルシーケンスを表したものである。呼び出し側のオブジェクト(図の左側)はメッセージパッシングをすると実行を中断し、停止状態になる。それ自身オブジェクトであるメッセージはＳＴＵＢでエンコードされ(図の白丸)、ＲＰＣ、ＲＢＴの各層を介して伝達され、ＳＴＵＢでデコードされて(図の黒丸)呼び出されたオブジェクト(図の右側)に渡され、実行の準備が整うと呼び出し側に先ほどと逆の経路で確認が返される。渡されるべきオブジェクトが何らかの理由で存在しないときには、エラーメッセージが確認として返される。確認を返された呼び出し側オブジェクトは停止状態から起こされて、確認の内容を調べ正常であれば結果が返ってくるのを待つ。エラーが有れば処理ルーチンが起動される。このメカニズムにより、オブジェクトの呼び出しに失敗したときに生じる呼び出し側オブジェクトの永久停止(デッドロック)を回避することが出来、またエラーの処理をユーザがプログラムすることを可能にしている。

メッセージパッシングは、ＯＺ+の分散処理の最も基本となる機能であり、メッセージパッシングによりデータであるオブジェクトが変更されてゆき実行が進んで行く。

3) 測定と解析

測定は、計算機性能による影響を排除するために全てのステーションにSparc Station 2を用いた。複数のステーション、プロセスにまたがる測定のために、ＣＰＵ時間ではなく、経過時間をシステムコールのftimeを用いて測定を行い、一万回の繰返しの平均値を求めた。

図3に、オブジェクトの位置の違いによるメッセージパッシングの3つの場合を示している。Case Aは同じVM内、Case Bは同じステーションの異なるVM、

An Evaluation of the Object Communication mechanisms of OZ+.
Yoichi Hamazaki and Michiharu Tsukamoto
ElectroTechnical Laboratory

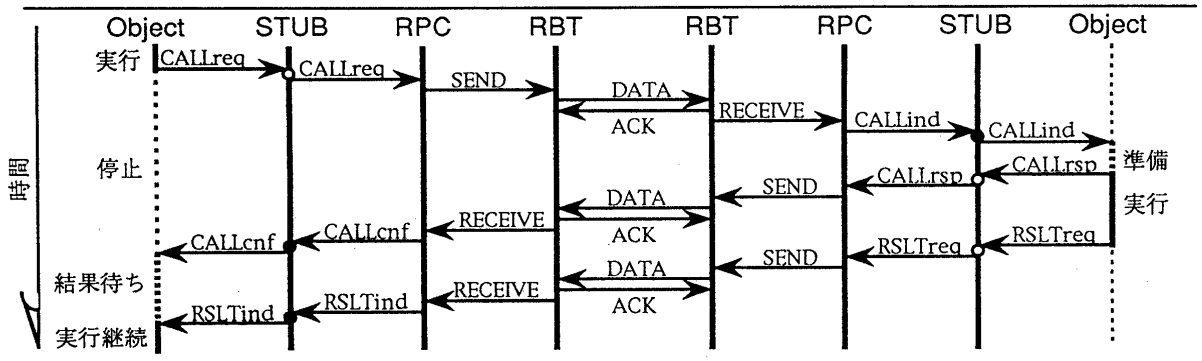

図2 メッセージパッシングのプロトコルシーケンス(正常時)

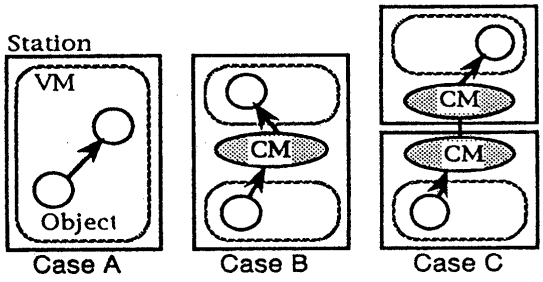

図3 二つのオブジェクトの位置

表1 通信に要する時間(m s)

	Case A	Case B	Case C
メッセージパッシング	0.364	198.4	213.2
R P C		159.0	188.8

Case Cは異なるステーションの場合である。メッセージパッシングでは、すぐに値を返すようなメソッドを呼び出した場合を、ＲＰＣではなにも処理をしないプログラムを上位テスタとした場合を測定した。それぞれの場合のメッセージパッシングおよびＲＰＣに要する時間の測定値を表1に示す。

メッセージパッシングのCase Aの場合では、オブジェクトの受渡しは直接ポインタをやり取りすることで行われてオブジェクト通信は起こらないので、case Aの測定値は、ＢＣＩにおける処理時間と見なせる。ＲＰＣに要する時間とメッセージパッシングに要する時間の差が、ＳＴＵＢでのオーバーヘッドと考えられる。ＳＴＵＢではデコード/エンコードがそれぞれ3回ずつ起き、そのオーバーヘッドは1組のデコード/エンコードで約13msである。Case BとCase Cで差が同じにならないのは、Case Cでは異なるステーションで実行が並行に行われるために、処理時間が隠れている為と思われる。

RBT間の通信では、データ送信要求からデータ確認(ＡＣＫ)が戻ってくるまでの時間を様々なデータ長nについて測定した。

同じステーション内では、21.2+0.00062 n ミリ秒

ステーション間(Ethernet経由)では、

48.8+0.002 n +int((n-1)/6144)*5.3+int((n-1)/12288)*7.3

ミリ秒かかる。ステーション間の時間の第3項は最大6kBのデータパケットに分割されるために、第4項はウィンドウサイズを2にしたために生じた項である。ステーション内では、メモリ間のコピーによりデータが転送されるので、こうした項は生じない。

ＲＢＴでのデータ転送のために、送受信それぞれに順序機械を生成して制御を行っているが、そうした機構のオーバーヘッドがステーション内の場合の定数項21.2msに相当し、ステーション間のデータ転送にかかるオーバーヘッドが25ms程度と推定される。

Case BのＲＰＣに要する時間について分析する。ＲＰＣ間の通信では、ＣＭを介して二つのプロセスが通信する。プロセス間の通信は、共有メモリ上にソフトウェアで実現されたFIFOを用いて行っており、1回で14.4ms要する。プロセス間の通信は図2から分かるようにＲＰＣのシーケンスの中で6回起こるので、そのためのオーバーヘッドは86.4msとなりＲＰＣ全体の54%を占めている。ＲＢＴによる通信は3回で、約64msでＲＰＣ全体の40%を占める。残りの6%がＲＰＣ層でのオーバーヘッドである。

Case BとCase CでのＲＰＣの時間の差は、ＲＢＴのステーション内とステーション間の場合の差の1.5倍程度となっているが、これはCase Cでは処理が異なるステーションで並行に行われるためである。

Case Bのメッセージパッシングについてオーバーヘッドの割合を多い順に示すと、プロセス間通信に43.5%、ＲＢＴで32%、ＳＴＵＢで20%、ＲＰＣで4。3%、そして実際のメソッドの呼び出しは0.2%である。

4) まとめ

ＯＺ+の通信機構に関して、その実装と各レベルでの性能の経過時間に基づく測定/評価について述べた。異なる仮想マシン間の通信では、プロセス間通信がオーバーヘッドの大きな要因になっており、ＲＢＴのオーバーヘッドと共に改善すべき点である。

ＯＺ+は純粋にオブジェクトを指向し、オブジェクトをオブジェクトとして通信することができ、またその行き先の位置にかかわらず統一的な扱いが出来る。そのために、ＵＮＩＸのＲＰＣのような単純な通信に比べるとオーバーヘッドは大きくなっている。

今後、この評価に基づいて、サーバーなどのシステム管理機能などシステム全体の評価を進めると共に、新しいシステムを設計する上で重要な基礎データとして活用する方針である。

ＯＺ+の開発は、工業技術院大型プロジェクト「電子計算機相互運用データベースシステム」においてなされたものである。

参考文献

[1] 塚本、濱崎、他:"オブジェクト指向開放型分散システムＯＺ+の開発"、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」研究成果発表会予稿、1991.11

[2] Tsukamoto,M , Sato,Y. et. al.:"The Architecture of OZ: Object-Oriented Open Distributed System", Proc. 2nd International Symposium on Interoperable Information Systems, 1988.11

[3] Goldberg,A. and Robson,D.:"Smalltalk-80: The Languages and its Implementation",Addison-Wesley,1983