

## 5 G-2

適応的にデータ表現変換を行なう  
リモート・プロシジャ・コールの検討

加藤 聡彦      藤長 昌彦      杉山 敬三

国際電信電話(株) 上福岡研究所

## 1. はじめに

複数の計算機をネットワークで結んだ分散システムにおいて、クライアント・サーバモデルに基づくリモート・プロシジャ・コール(RPC)が広く利用されている。異なるハードウェアやオペレーティング・システム(OS)が混在する異機種計算機環境においてRPCを行なうためには、何らかの方法でデータの表現形式を変換する必要がある。現在の分散システムは、ネットワーク標準のデータ表現形式を定め、データ表現の多様性に対処している<sup>[1,2]</sup>。標準形式と異なる内部表現を持つ計算機がネットワーク通信を行なう場合、内部表現から標準形式へ表現変換を行なうことで、異機種間通信を行なうことができる。

しかし、この手法では計算機の内部表現と標準形式とが異なる場合、本来なら変換の必要がない同機種間通信であっても表現変換を2度行なうことになる。これはネットワークを介したRPCにおいて、通信相手の内部表現の形式を認識しないためである。そこで、通信相手を認識し、

- ① 不要な表現変換を行なわない
- ② 表現変換が必要な場合にも変換コストを最小とする

ことを目的とする「適応的なデータ表現変換機能」をRPCに持たせることにより、ネットワークを介したRPCを高速化することが可能である。

通常、RPCではクライアントがサーバコールに先立ちネームサーバにアクセスし、クライアントとサーバの論理的な結合を行う<sup>[3]</sup>。そこで、ネームサーバにデータ表現形式に関する情報を持たせることにより、データ表現形式に関するネゴシエーションを行なわせることができる。

本稿では、適応的データ表現変換機能をもつRPCの提案とその実現手法について述べる。

## 2. 適応的なデータ表現変換の提案

## 2.1 標準データ表現形式への変換コスト

異機種計算機間のRPCにおいて標準形式を採用することの利点は、実装が容易となる点である。各計算機は標準形式への1通りの変換を実装すればよく、ネットワーク上の他の計算機に関する知識あるいは通信開始時の上位の手続きを必要としない。これは不要な変換を伴う場合があるが、イーサネット/IP/TCPあるいはUDPを用いた現在のRPCにおいては、ネットワーク通信そのもののコストが高いため、データ表現の変換に要するコストは無視できると考えられてきた<sup>[2]</sup>。

しかし、筆者らの簡単な実測によれば、Sun3/260において、8KB整数データのバイト順序変換に10msec程度を要する。このオーバーヘッドは高速な分散処理を目指す新しい分散システムにおいては問題となり得る。例えばRPCに適した新しいトランスポートプロトコルを採用したVシステムでは、8KB整数データをイーサネット上において17.8msecで転送できたという報告がなされている<sup>[4]</sup>。さらに、光ファイバを伝送路とする高速ネットワークの普及が進むにつれ、データ表現変換に要するコストがますます無視できなくなるものと予想される。

## 2.2 適応的データ表現変換の機能

本稿で提案する適応的データ表現変換では、同機種間通信の速度を損ねることなく、異機種間通信を可能とするために、次のような方法を用いる。

- ① 両者を判別し、
  - (a) 同機種間の場合  
表現変換を一切行なわない。
  - (b) 異機種間の場合  
変換が必要なデータ型のみ表現変換を行なう。
- ② 表現変換を行なう場合には、
  - (b.1) 一方が他方の内部表現に変換する。
  - (b.2) 両方の計算機が標準形式に変換する。

二種類の変換方法を用いたのは、前者が可能な場合には、処理能力や動的に変化する負荷、変換に要するコスト等を考慮しながら、適応的に表現変換を行なう計算機を選択することでより効率の高いRPCが実現でき、また標準形式を準備しておくことで、特殊な内部表現を持つ計算機や非常に複雑なデータ構造を扱う場合の通信性も確保できるためである。

### 3. 適応的なデータ表現変換の実現手法

#### 3.1 表現変換方法の選択

前述のように、ネームサーバにデータ表現変換に関するネゴシエーション機能を持たせ、適応的に表現変換を行なうRPCを実現する。このRPCの手順を図1に示す。まず、サーバは自身のサーバ名とサーバ・ポートをネームサーバに登録する(signIn)。クライアントはサーバ名を引数として、サーバ・ポートをローカルホストのネームサーバに問い合わせる(lookUp)。サーバが他の計算機上に存在する時、サーバ名の問い合わせとともに、内部表現形式、負荷、処理能力に関する情報を交換し、表現変換が必要な場合には、最適なデータ表現変換を適応的に選択する。サーバ・ポートとともに、選択した変換情報をクライアントに通知する。またサーバに対しても変換情報を通知する。その後クライアントとサーバは適応的なデータ表現変換を行いつつ、RPCを実行する。

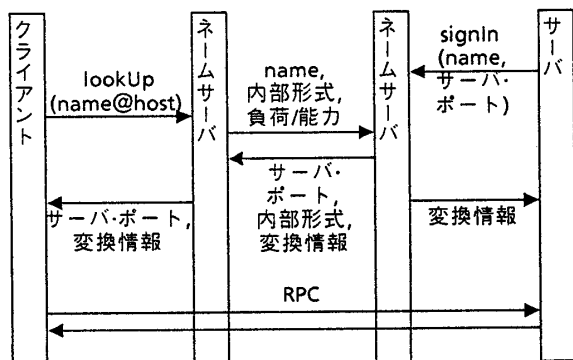


図1 適応的なデータ表現変換機能をもつRPCの手順

#### 3.2 スタブによる表現変換

一般にデータ表現の変換を行なう方法としては、ネットワーク通信のサーバ(ネットワーク・サーバ)を導入しそれが行なうもの<sup>[1]</sup>と、スタブが行なうもの<sup>[2]</sup>が考えられる。

適応的なデータ表現変換機能では、内部的なデータ表現をネットワーク上のデータ表現とするために、データ自身を見てもそのデータ型を判定することはできない。ネットワーク・サーバは、RPCに関係するクライアントとサーバと独立して設けら

れるため、すべてのRPCの仕様を知ることは困難である。従って、ネットワーク・サーバを用いる方法では、整数や実数等の基本データ型の変換しかサポートできない。複雑なデータ構造を引数に持つRPCを使用する場合、プログラマが陽にネットワーク標準形式への変換関数を用意しなければならない。

表現変換をRPCスタブにおいて行なうことにより、この問題を解決することができる。スタブはRPCの引数に関する知識を利用できるため、複雑なデータ構造を引数に持つRPCが実現でき、更に、RPC引数のデータ型ごとにきめの細かい適応的なデータ表現変換を行なうことも可能となる。

#### 3.3 実現上の検討項目

本手法を実現するにあたり、以下の項目について詳細に検討する必要がある。

##### (1) データ表現形式

スタブに組み込むべき代表的な内部表現形式の抽出、および低コストに表現変換が可能な標準形式の設計。

##### (2) ネゴシエーションのパラメータ

ネームサーバ間で交換するパラメータの抽出と最適な変換法を選択するアルゴリズム。

##### (3) ネゴシエーションのコスト

ネゴシエーションおよびデータ表現変換を含めた、全体としてのコストを最小とするネゴシエーションの手法。

#### 4. むすび

本稿では、異機種計算機環境におけるRPCを高速度化するために、必要な場合にのみデータ表現の変換を行なう適応的なデータ表現変換機能を持つRPCについて述べた。今後スタブジェネレータの作成を含め、複数のOSが混在するネットワーク上に分散システムを構築するための基礎検討を行なってゆく。最後に日頃御指導頂くKDD上福岡研究所小野所長、浦野次長、鈴木コンピュータ通信研究室長に感謝する。

#### 文献

- [1] Tevanian, Rashid, "MACH: A Basis for Future UNIX Development", CMU-CS-87-139, June, 1987.
- [2] Sun Microsystems, Inc., "Network Programming", 1988.
- [3] Birril, Nelson, "Implementing Remote Procedure Calls", ACM Trans. on Comp. Syst., Feb. 1984.
- [4] Cheriton, "The V Distributed System", Communications of the ACM, March, 1988.