

異機種分散処理環境におけるインタフェース自動生成処理方式

7Q-3

(2) インタフェースプログラム生成方式

関根徹 末広亮太 田中伸明 榎本典行 本田邦夫

松下電器産業(株) 情報通信東京研究所

1. はじめに

我々は、ネットワーク上に接続された異機種計算機間で、相互に通信及びデータ交換するための方法としてRPCを採用し、分散処理環境におけるアプリケーションプログラム(AP)の開発を容易にするため、インタフェース(I/F)自動生成処理方式の検討を行った。本稿ではこのRPCを実現し、異機種性を吸収するI/Fプログラムを自動的に生成する方式について説明する。

2. RPC利用における相互接続上の問題点と解決方法

分散環境においてRPC利用の相互接続上の問題点には、プログラム記述言語、データ型表現などの言語仕様に依存するものと、バイトオーダ、ワードサイズ、浮動小数点の表現方法、またバウンダリ調整などの機種に依存するものの2つに大きく分類できる。前者は生成するI/Fプログラムコードに関係し、後者は実行時の転送データ表現に関係する。さらに、I/Fプログラムを生成する計算機や、生成する言語種別も考慮する必要がある。

2.1 交換データのデータ型

任意のプログラム間でデータ交換を行うためには、その交換するデータ型がI/Fとして適合していなければならない。既存のプログラミング言語でこのI/Fを記述することは、言語間に仕様の違いがあり、そのままデータ交換を行うことは困難である。そこで我々は各言語間の共通の仕様を備えたI/F定義言語を設計した。ユーザはこの言語を使用しI/F部分を記述することで、交換データのデータ型の一般化が可能になる。また、この一般化を行うためには、相互のユーザAP間の言語仕様の差異を吸収する必要がある。ネットワーク上の異機種計算機の言語仕様の差異をデータベースとしてファイル化し、I/Fプログラムソースコード自動生成時にこれを参照/照合することでこの問題を解決した。

2.2 転送データ

異なる計算機間でデータ交換を行うためには、相互の計算機でそのデータを解釈できなければならない。そこで一般的に用いられているアプローチ、つまり転

送データ表現の一般化を行った。ここで考慮すべきことは、ローカルな一般化ではネットワークを外部に拡張した際には真の共通化にはならないということで、一般化のための共通データ仕様にOSIのASN.1フォーマットを採用した。また、異なる計算機がそれぞれ転送データ表現を一般化する方法として、実行時にデータのエンコード/デコードを行うライブラリを計算機単位に用意する方式を採用した。

2.3 I/Fプログラムソースコード

I/Fプログラムソースコードをどの計算機上で、どの記述言語で生成するかが問題になる。生成する計算機に依存するソースコードであれば、RPCを実行する計

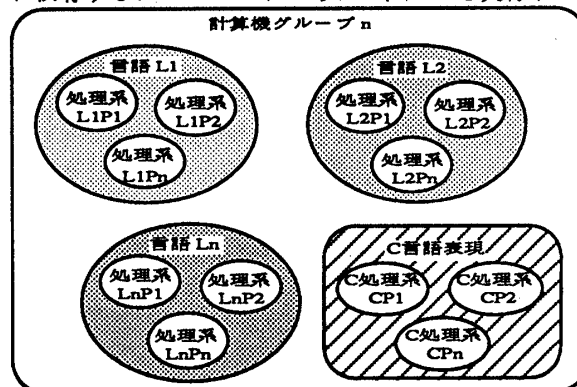


図1 言語情報ファイルの概念的構成

算機の種類だけ必要となる可能性もあり、生成する記述言語によってはRPCを実行する計算機がサポート不可の場合がある。そこで、生成する記述言語はシステム記述言語として一般的であり、分散環境で広く使用されているC言語を採用した。さらに、言語情報管理にネットワーク上の異機種計算機のC言語記述表現を含むことで、I/Fプログラムソースコードを生成する計算機が、RPCを実行する計算機と異なった場合でも問題なく動作することを可能とした。

3. I/Fプログラム自動生成方式

次にI/Fプログラム自動生成方式のうち、特徴ある部分を説明する。

An Automatic Interface Generator for Distributed Processing Systems in Heterogeneous Environment

(2) Generating Mechanism of Sources for the Interface Program

Toru SEKINE, Ryota SUEHIRO, Nobuaki TANAKA, Noriyuki ENOMOTO, Kunio HONDA

Matsushita Electric Industrial Co., Ltd.

3.1 言語情報管理

言語情報ファイルは図1に示すように、ネットワーク上の異機種計算機に関して、同種の言語仕様及びC言語表現を持つグループ単位に構成される。そして、個々のグループは次のようなカテゴリに分類される。

- a) 言語種別 … 使用可能言語(C, Pascal, Fortran, …)
 - b) 処理系種別 … 言語種別毎の処理系
 - c) C言語表現種別 … C言語ソースコード表現
 - d) C言語処理系種別 … C言語表現種別毎の処理系
- そして、実際の言語情報はb)及びd)の構成要素として存在し、特にb)の言語仕様には以下のものがある。

- ・ 識別子区別 … 大文字/小文字
- ・ 識別子長 … 最大文字数
- ・ RPC引数タイプ … Call-by-Value/Call-by-Reference
- ・ サポートデータ型 … 以下参照
- ・ 配列順序 … 多次元配列の場合の順序

サポートデータ型には、単純データである符号付整数型、符号なし整数型、実数型、オクテット列型、ブーリアン型、ポインタ型と、構造を持つデータである配列型、構造体型がある。本生成方式は、上記a), b), c), d)をキーワードに、相互接続を行う計算機間の言語情報をデータベースから読み込み、次の交換データのデータ型の生成、I/Fプログラム生成、及びユーザ定義データ型のエンコード/デコードに利用する。

3.2 交換データのデータ型生成

ユーザがI/F定義言語で記述したクライアント/サーバ両計算機の交換データのデータ型を、各々のC言語記述表現に適合するデータ型定義として生成する。これは、I/F定義の構文を解析し解析木として分解した各要素を、両計算機の言語情報と照合した上で、C言語の単純データ型、ポインタ型、配列型、もしくは構造体型のいずれかの新規データ型定義(`typedef`)を行い、ヘッダとして生成する。

3.3 インタフェースプログラム生成

クライアント及びサーバのI/Fプログラムソースコードは、いずれもRPC関数定義部、通信I/Fライブラリ呼び出し部、エンコード/デコードライブラリ呼び出し部、そしてユーザAP間I/F部からなる。RPC関数定義部は関数の引数と対応するI/F定義のデータ型をマッピングし、通信I/Fライブラリ呼び出し部はネットワークに適合したプロトコル(TCP/IP, UDP/IP)を実現するライブラリを取り込む。エンコード/デコードライブラリ呼び出し部は、機種に応じた関数の引数/結果の転送データ表現を一般化するライブラリを取り込む。そしてユーザAP間I/F部は、クライアントならばユーザAPがI/Fプログラムを呼び出す形式で、サーバならばI/FプログラムがユーザAPを呼び出す形式で生成される。

3.4 ユーザ定義データ型のエンコード/デコード

ユーザが記述するI/F定義には、システムが提供し言語情報のサポートデータ型の単純データに相当する予約データ型と、同サポートデータ型の構造を持つデータに相当し、ユーザ自らが予約データ型を組み合わせる新たなデータ型を定義したユーザ定義データ型とがある。一方、エンコード/デコードライブラリは予約データ型対応のみが存在する。そこでシステムは、このユーザ定義データ型のエンコード/デコードを行うため、図2に示すように、I/F定義を参照して予約データ型のエンコード/デコードライブラリ呼び出しに分解する。

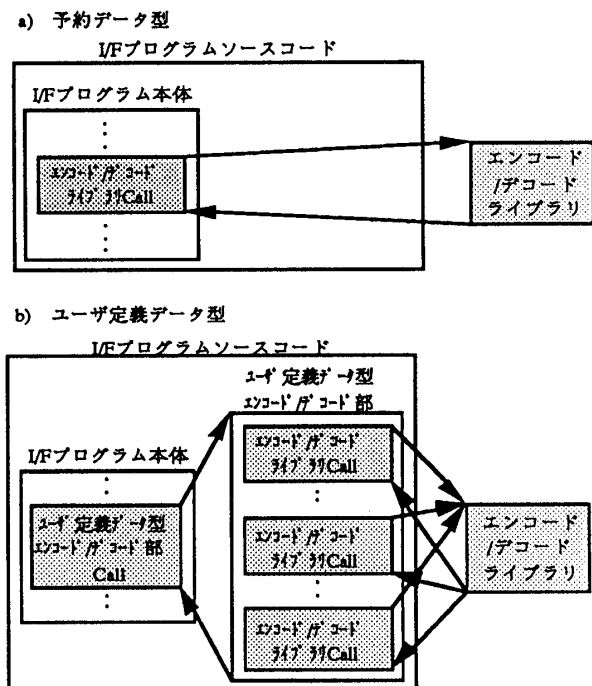


図2 エンコード/デコードソースコードの生成

4. まとめ

我々は、I/F定義により交換データを、エンコード/デコードにより転送データの一般化を行い、さらにI/Fプログラムをソースコードとして生成することで異機種性を吸収した。

なお、本研究開発は、通商産業省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の研究開発」の一環として、新エネルギー・産業技術総合開発機構(NEDO)より委託を受けて実施したものである。

参考文献

- [1] Phillip B. Gibbons, "A Stub Generator for Multilanguage RPC in Heterogeneous Environment", IEEE Trans. Soft. Engineering, Vol. SE-13, No.1, Jan. 1987
- [2] Network Computing System (NCS) Reference 10200, Apollo Inc., MA01824, Jun. 1987