

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-2

(1) 基本処理方式

末広亮太 関根徹 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

コンピュータネットワークの急速な発展に伴い、異機種のコンピュータが、同一ネットワーク内に多数共存するようになってきたが、これら異機種のコンピュータが相互運用性を持った分散環境として使用されるにはまだ至っていない。

本研究の目的は、異機種ネットワーク環境において、計算機の相互運用を容易に実現するため、言語/機種の違いや、ネットワークとのインタフェースを意識することなく、ユーザが分散アプリケーションプログラムを開発できるようにすることである。

本稿では、上記の問題を解決するためのRPC (Remote Procedure Call: 遠隔手続き呼び出し)[1][2]を用いたインタフェース自動生成[3][4]に関する方式検討を行ったので、その全体概要について報告し、[5]では、インタフェースプログラムの生成方式について、[6]では、データのエンコード/デコード方式について報告する。

2. システムの要求仕様

本システムは、クライアント/サーバモデルの分散アプリケーションプログラムを想定しており、アプリケーションプログラムからの要求メッセージの言語/機種の違いやネットワークとのインタフェースなどは、本システムによって生成されるインタフェースプログラムとリンクすることによって、吸収/隠蔽される。

言語/機種の違いとは、プロセッサ、文字コード、アプリケーション言語、ソフトウェアライブラリの違いなどを意味し、これらを吸収するために、アプリケーションの呼び出し形式を定義するRPCインタフェース定義言語の仕様を明確に決め、また、ネットワーク上の実装においては、相互運用性を考慮し、データは、ASN.1エンコーディングルールに従うものとした。

3. システム構成

3.1 実行時のソフトウェア構造

実行時のソフトウェア構造を図1に示す。クライアント/サーバはそれぞれ1つのプロセスとして動作し、クライアントアプリケーションからの要求は、エンコーダでASN.1転送構文に変換されてサーバに送信され、サーバにおいてサーバアプリケーションプログラムに適したデコード処理が行われる。結果データについても同様の処理が行われ、クライアントアプリケーションに結果データが返される。

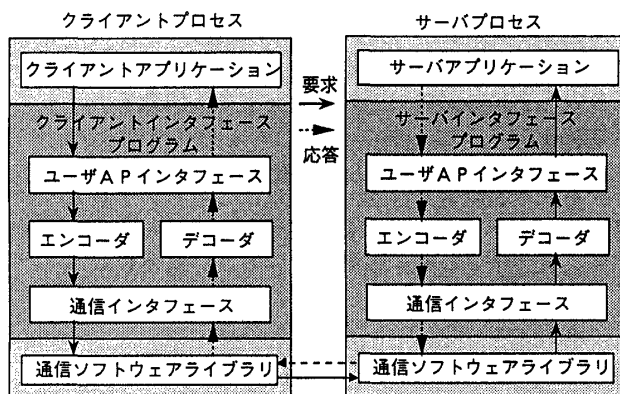


図1 実行時のソフトウェア構造

3.2 インタフェースプログラムの生成方式

インタフェースプログラムの生成方式を図2に示す。まず、図2(1)に示すように、ユーザの作成したRPCインタフェース定義、RPC環境情報定義の2つのファイルを入力情報とし、インタフェースプログラムジェネレータによって、言語情報データベースファイルを参照することにより、C言語のヘッダと、クライアント/サーバそれぞれのインタフェースプログラムソースを生成する。

・RPCインタフェース定義 (ユーザ定義)

インタフェース定義言語を使用して、遠隔手続き呼び出しのインタフェース定義を行う。このRPCインタフェース定義言語によって、遠隔手続き呼び出しを記述した例を図2(3)に示す。

・RPC環境情報定義 (ユーザ定義)

クライアント/サーバの機種名、アドレス、プロトコル、アプリケーションプログラムの言語名などの定義を行う。

・言語情報 (システム保持)

機種に依存した言語の仕様などの情報が格納されているデータベースファイル。

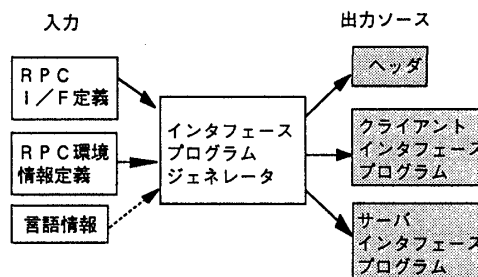


図2(1) ソースプログラム生成

An Automatic Interface Generator for Distributed Processing Systems in Heterogeneous Environment

(1) System Architecture

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

Matsushita Electric Industrial Co., Ltd

次に、図2(2)に示すように、生成されたインタフェースプログラムソースを、クライアント/サーバマシンにそれぞれ転送し、マシン上のエンコード/デコードライブラリ関数、通信インタフェースライブラリ関数、そしてユーザの作成したクライアント/サーバアプリケーションプログラムとリンクしてクライアント/サーバプログラムオブジェクトを生成する。

・エンコード/デコードライブラリ関数

インタフェース定義で定義された個々のデータをASN. 1にエンコード/デコードする関数のライブラリ。

・通信インタフェースライブラリ関数

データの送受信を行う。現在は、TCP/IPとUDP/IPプロトコルをサポートしている。サーバは、デーモンとして起動されており、要求に応じてプロセスを作成する。

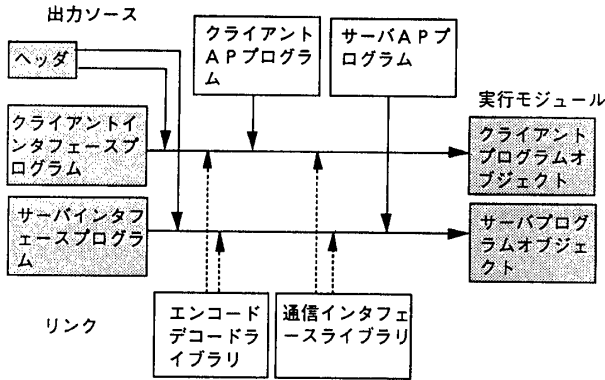


図2(2) 実行モジュール生成

```
rpc float_32 remote_func(
    arg1:unsigned_int_16:in;
    arg2:float_32:out;
    arg3: float_32:inout);
```

図2(3)RPCインタフェース定義

3.3 インタフェースプログラムジェネレータ

インタフェースプログラムジェネレータの構造を図3に示す。構成要素の処理内容の説明を以下に示す。

・インタフェース定義言語解析部

アプリケーションプログラム間の遠隔手続き呼び出しのインタフェースを定義したRPCインタフェース定義を入力し、インタフェース定義言語仕様に基いて解析し、その情報を解析木として生成する。

・データ構造管理部

インタフェース定義言語解析部が生成した解析木を管理し、インタフェースプログラム生成部がインタフェースプログラムを生成するとき、その要求に応じて必要な情報を提供する。

・インタフェースプログラム生成部

RPC環境情報と解析木を入力とし、言語情報を参照してクライアントとサーバの機種に応じたC言語のインタフェースプログラムのソースコードを生成する。

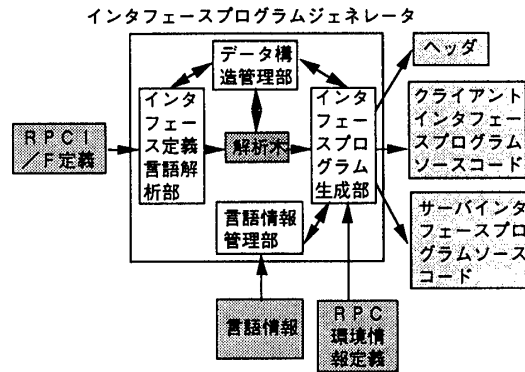


図3 インタフェースプログラムジェネレータ

・言語情報管理部

インタフェースプログラム生成部の要求に応じて、言語情報ファイルから必要な情報を提供する。

5. まとめ

以上述べたように、分散アプリケーションのためのインタフェースを自動生成することにより、分散アプリケーションの開発効率が上がり、また、言語/機種/ネットワークを意識しない開発が可能となると考えられる。

今後、実験システムの構築、評価、及びRPC I/F定義、RPC環境情報定義の機能面/使用面での評価などを行って行く予定である。

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] Andrew D. Birrel, Bruce Jay Nelson, "Implementing Remote Procedure Call", ACM Transactions on Computer Systems, Vol.2, No.1, Feb.1984
 [2] Remote Procedure Call Protocol Specification, Sun Microsystems, Inc., 2550 Garcia Avenue Mountain View, CA 94043, 1986
 [3] Michael B. Jones, Richard F. Rashid, Mary R. Thompson, "Matchmaker: An Interface Specification Language for Distributed Processing", proceedings of the 12th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Language, ACM, pp225-235, Jan, 1985
 [4] 榎本 他: "異機種環境における遠隔手続き呼び出し用言語インタフェースの生成", 電子計算機相互運用データベースシステムの研究開発中間成果発表会, Nov.30-Dec.1989
 [5] 関根 他: "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 (2)インタフェースプログラム生成方式", 情報処理学会第41回全国大会講演論文集
 [6] 田中 他: "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 (3)エンコード、デコード方式", 情報処理学会第41回全国大会講演論文集