

## サン・マイクロシステムズの考える分散処理環境

樋口貴章

日本サン・マイクロシステムズ株式会社

オープン・ネットワーク・コンピューティング (ONC(tm)) のリモート・プロシージャ・コール (RPC) は、分散処理アプリケーションがお互いに通信を行うための高レベルのパラダイムを提供している。当初のRPCは、UDP/IPとTCP/IPという特定のプロトコルの上に実装されていた。現在では、特定のトランスポート層に依存しない実装を行うことによって、アプリケーションが新しいトランスポート層にも容易に対応できるようになっている。また、RPCの上にさらに複雑な分散処理アプリケーションの通信を可能とする ToolTalk サービスが提供されており、分散処理環境にオブジェクト指向化の方向も取り入れつつある。ToolTalkは、今後のネットワーク分散オブジェクト環境の中核となるオブジェクト・リクエスト・ブローカー (ORB) の実装に向けての掛け橋としての役割を果たすことになろう。

## Distributed Computing of Sun Microsystems

Taka Aki Higuchi

Nihon Sun Microsystems K.K

Remote Procedure Call(RPC) of Open Network Computing(ONC) provides a high level paradigm which let distributed applications communicate each other. Original RPC was implemented on specific protocols called UDP/IP and TCP/IP. Now RPC implementation does not depend on specific protocols, so that distributed applications can easily adopt new transport protocols. And now ToolTalk service, which is implemented on Transport Independent RPC and enables more complicated communications among distributed applications, is provided, it is intended to support object oriented environment in distributed computing. ToolTalk is positioned as the predecessor to ORB(Object Request Broker) implementation, which will be a core technology of the future distributed object environment.

## 1. まえがき

本稿では、ONCの歴史と現状を中心に説き、最後に将来の展望にも触れ、サンの分散処理環境に対する取り組みを考える。本稿を作成するに当たっては、サン・マイクロシステムズ株式会社のシステムソフトウェア開発子会社であるサンソフト社から提供されている技術白書を参考としている。技術的な詳細に関しては、文献リストの参考資料を参照していただきたい。

## 2. ONC RPCの歴史

元々のONC RPCは、UDP/IPとTCP/IPのトランスポート層の上に実装されていた。トランスポート層とのインタフェースには、4.2BSDのsocketインタフェースを使用していた。RPCの設計段階においては、特定のトランスポート層に依存するような設計ではなかったが、実際に実装を行っていくに当たって、TCP/IP固有の機能を使用している部分があることが明らかになった。これは主にネーミングとバインディングに関する実装部分である。socketは、ある程度はトランスポート層に対する独立性を打ち出すことに成功したが、あらゆるトランスポート層を容易に切り替えるという仕組みにはなっていない。このため、トランスポート層から独立したRPCの要求が出てくることとなった。

## 3. トランスポート・インディペンデントRPC(TI-RPC)[1]

今日のネットワーク・トランスポートは、大きく分けてデータグラムとバーチャル・サーキットの2つに分けられる。データグラムでは、転送のオーバーヘッドは少ないが、パケットの転送は保証されていない。パケットは失われるかもしれないし、順番がくるって到着するかもしれない。バーチャル・サーキットでは、ある程度のオーバーヘッドはあるが、転送は保証されている。この2種類に大別されるトランスポート層を使用することを前提にRPCの設計を行えば、トランスポート層に依存しないRPCの設計が可能となる。

このために、RPCのネーミングとバインディングの実装方式を変更し、トランスポート層に依存しないようにしたが、RPC自体は元々トランスポート・インディペンデントであり、インタフェースの変更は不要であった。

トランスポート・インディペンデントRPCは、ネットワークの選択・ネームからアドレスへの変換・トランスポート層インタフェース/ストリームの3つのモジュール構成で実装されている。ネットワーク選択モジュールでは、トランスポート層を動的に選択する仕組みを提供する。ネームからアドレスへの変換モジュールでは、トランスポート層のアドレスが統合された参照方法で得られる仕組みを提供する。トランスポート・レイヤー・インタフェース(TLI)では、RPCライブラリがどんなトランスポート層とも通信可能な仕組みを提供する。

トランスポート・インディペンデントRPCは、RPCのプロトコルを変更することなく実現されている。変更点の多くはクライアントとサーバのハンドルを生成する部分で行われており、それ以外の部分に関してはインタフェースの変更を行っていない

い。よって、トランスポート・インディペンデントRPCは既存のRPCとの互換性がある。

トランスポート層からRPCを独立させるメリットとしては、具体的には既存のTCP/IPベースのネットワークから、OSIに移行するという場合に、スムーズにシステムの移行が行えるというようなことが上げられるであろう。

#### 4. ONC RPCの上の各サービス

ONC RPCの上には、数多くのネットワークサービスが構築されているが、ここでSunOSに標準機能として搭載されているサービスに関して触れることにする。

##### 1) XDR(eXternal Data Representation)

XDRは、OSI参照モデルでのプレゼンテーション層に位置し、ネットワーク上での機種独立のデータを記述するための役割がある。XDRを使用することによって、データ・バイトの並び、データ型のサイズ、形式、整列の違いをなくし、データの交換を行うことができる。

##### 2) NFS(Network File System)

NFSは、ネットワーク上のファイルもローカルなファイルと同様に扱うことを可能とするサービスで、ユーザに対して透過的なものとなっている。NFSは移植性も高く、現在ではUNIX以外の様々なオペレーティング・システムの上に移植されており、その対象となるハードウェアもPCからメインフレームまで幅広く受け入れられている。

##### 3) NIS(Network Information System)

NISは、ネットワーク上の資源を共通

アプリケーション・プログラム

NFS	NIS	REX	Auto-mount	NLM
-----	-----	-----	------------	-----

RPC

XDR

トランスポート・レベル・インタフェース  
(TLI)

低レベル・ネットワーク・プロトコル  
(TCP/IP, OSI, SPX/IPX)

に管理するためのネーミング・サービスで、ユーザ管理・ホスト管理・エイリアス管理などをネットワーク共通に行うことができる。

##### 4) REX(Remote EXecution)

REXはリモート・システムの上でユーザ・コマンドを実行するためのサービスであるが、同様のサービスを提供するBSD系のコマンドrshとの違いは、リモートでシェルを起動しないために処理速度が速いことと環境変数はローカルのもがそのままコピーされて使用されるという点にある。

##### 5) NLM(Network Lock Manager)

NLMは、System VとMS-DOSのファイルに対して、ネットワークを介したレコード・ロック機能を提供する。これによって、PC-NFSによって使用されているファイルシステム上でもレコード・ロックを行うことが可能となる。

##### 6) Automounter

automounterは、必要に応じてファイ

ル・システムのマウントを行うサービスである。ユーザが使用しないファイル・システムは自動的にアンマウントされ、必要なファイル・システムは自動的にマウントされる。この機能によって、ユーザはネットワーク上のどのシステムを使用する場合でも同じ環境を使用することが容易となる。

## 5. NIS+[2]

ネットワーク環境の普及によって、システム管理者・エンドユーザ・アプリケーション開発者のそれぞれが新たな問題に直面することになる。サブネットワークが複数接続された大規模な分散ネットワーク環境においてそれは典型的なものとなるであろう。ネットワークが大規模になればなるほど、ネットワーク上の資源に透過的にアクセスするという複雑な要求が強くなってくるのである。このような大規模な資源の分散共有を行うサービスはエンタープライズ・ネーミング・サービスと一般的に言われている。NIS+(Network Information Service Plus)は、そのようなマルチ・ベンダの分散処理環境にたいしてのエンタープライズ・ネーミング・サービスの一つとしてサンが提供するものである。

NIS+では、NISに対して、3つの拡張がなされている。

- 1) グローバル・ネームスペース構造
- 2) マップ内のデータ構造
- 3) 認証サービス

さらに、今までNISの上で使用されてきたアプリケーションや環境からの移行もスムーズに行えるように互換性も備えている。

## 6. ToolTalk[3][4][5]

RPCがアプリケーションとアプリケーションの1対1の通信を中心に考えられていることに対して、さらに複雑な分散処理アプリケーション作成のプラットフォームに対する要求がソフトウェア開発者から上がっている。アプリケーション間のデータの交換、操作の統合などを実現するサービスとして新たにRPCの上に築かれたのがToolTalkである。

ToolTalkでは、マルチキャスト機能によって、データを受け取る側のアプリケーションがネットワーク上のどこにあるかなどということは送り手側のアプリケーションが知らなくてもデータの転送を行うことが可能である。また、ToolTalkのAPIは既存のアプリケーションのソースコードに簡単な変更で追加できるようになっているため、複雑なポーティング作業を行う必要はない。実際に、昨年のCFIにおいては、マルチ・ベンダ環境でのToolTalkデモンストレーションを行ったが、これは既存アプリケーションに数行のToolTalk APIを実装することによって容易に実現できたのである。

ToolTalkの実際の使用形態としては、独立して開発されたアプリケーションが、相互に連携を行い、データのアップデートを送受信し、さらにはデータを元にアプリケーションの起動や終了を行うということが考えられる。実際にOpenWindows Version3のDeskSetと呼ぶアプリケーションの中ではToolTalkを用いてインターオペラビリティを実現している。

## 7. Project DOE[6]

分散処理環境でのアプリケーションの将来の形態はオブジェクト指向である。オブジェクトとしてのアプリケーションのメッセージのやりとりでインターオペラビリティを実現することになる。オブジェクト間のメッセージの受け渡しは、さらに複雑度を増したアプリケーション間通信となっていく。

この分散処理環境でのオブジェクトの取り扱いに関しては、OMG(Object ManagementGroup)において標準化作業が行われている。既にその中核となるORB(Object Request Broker)に関しては、サンとHPの共同提案であるDOMF(Distributed Object Management Facility)をベースに仕様が決定されている。

今後のサンの環境の中に、このDOMF(ORB)の機能を取り込んで、将来の分散処理環境を構築していくためのプロジェクトを、Project DOE(Distributed Object Everywhere)と呼んでいる。ToolTalkは、現在のTI RPCと将来の分散処理オブジェクト環境の掛け橋として位置付けられ、Project DOEではToolTalkに対する互換性を提供することになる。

## 8. まとめ

ONC RPCから始まったサンの分散処理環境の実装は、トランスポート・インディペンデントRPC、ToolTalkを経て、将来の分散処理オブジェクト環境であるORBまで見通して現在も進化しつつある。ここで重要なことは、既存技術に対する様々な投資

の蓄積を有効に活用しつつ、新しい技術への発展をはかる改革的アプローチである。分散処理環境でのオブジェクト指向の導入は、ハードウェア・アーキテクチャやオペレーティング・システムおよびGUIの違いを吸収し、完全にオープンなマルチ・ベンダー分散処理環境を実現することになる。

## 9. 文献リスト

- [1] McManis, C. & Samar, V., "Design and Implementation of Transport-Independent RPC", SunSoft, September 1991.
- [2] McManis, C. & Jang, S., "Network Information Service Plus(NIS+)", SunSoft, September 1991
- [3] Frankel, R., "Introduction to the ToolTalk Service", SunSoft, September 1991
- [4] Frankel, R., "The ToolTalk Service", SunSoft, September 1991
- [5] Frankel, R., "ToolTalk in Electronic Design Automation", SunSoft, September 1991
- [6] "Distributed Objects Everywhere", SunSoft, September 1991