マルチメディア通信と分散処理 93－6
（1999．4．16）

System Area Network における ORB 通信の高速化手法

石嵜 透，佐伯 敏章
今井 祐二，岸本 光弘

(株) 富士通研究所 マルチメディア研究所 ソフトウェア研究部

{tooru,saeki,kimai,kiss}@flab.fujitsu.co.jp

本論文では，System Area Network 上の ORB 通信オーバヘッド削減のために，従来の IIOP/TCP/IP に加え，System Area Network 上の低オーバヘッドなプロトコルを導入した Crisp ORB の提案を行う．導入したプロトコルは，下位レイヤに VI アーキテクチャ，上位レイヤに SIOP を使用したもので，従来に比べ，データコピー，システムコール，エラー制御などを削減している．System Area Network 外に対する通信は従来の IIOP/TCP/IP で行うことができ，処理の選択は ORB 内で自動的に行う．プロトタイプを設計・実装し双方向通信の性能評価テストを行った．導入したプロトコルによる短いメッセージの Round Trip Time は 414 μ s であり，従来の IIOP/TCP/IP に比べ 25%高速化された．

Smart strategy for upgrading ORB communication performance on System Area Network

Tooru Ishizaki, Toshiaki Saeki
Yuji Imai, Mitsuhiro Kishimoto

Software Laboratory Multimedia Systems Laboratories Fujitsu Laboratories Ltd.

{tooru,saeki,kimai,kiss}@flab.fujitsu.co.jp

In this paper, for the purpose of alleviating ORB communication overhead on System Area Network, we suggest Crisp ORB. Low overhead protocol has been introduced into Crisp ORB for replacing current IIOP/TCP/IP protocol on System Area Network. Introduced protocols are VI architecture as lower layer protocol and SIOP as higher layer protocol. Relatively to current protocol, reducing costs of data copy, system call overhead, error control and so on. Communicating to non System Area Network, we can use current IIOP/TCP/IP protocol. The selection of those protocols is processed within ORB. We made a prototype for Crisp ORB and measured 2-way communicaton performance. As a result, Round Trip Time with null string argument is 414 μ s. We cut off 25% of Round Trip Time.

1 序論

Electronic Commerce やイントラネットといったネットワーク上に多くの情報資産が分散する現在の情報サービス分野では，複数コンポーネントにより1つのシステムを構成するコンポーネントウェア技術が新たなプラットホームとして導入されている．コンポー

ネントは独立して開発・保守が行える物理的な位置が透過なオブジェクトの集合で，通信ミドルウェアを使ったオブジェクト間通信によりコンポーネント間の情報交換を行う．通信ミドルウェアはORB,RPC,MOMなど様々なモデルがあるが，ORBは標準アーキテクチャCORBA[1]により相互運用性を保証され，最も多くのベンダから製品が供給されている．

従来のORBはLANやWANで広く使われ，TCP/IP通信を行っていた．最近では，GigaBit Etherのような高速LANやSystem Area Networkといった高速通信ハードウェアが利用されるようになり，大量データの転送性能(bandwidth)がハードウェア性能に比例して向上するようになった．しかし短いメッセージでは，TCP/IPによる通信オーバヘッドでハードウェア性能を十分引き出すことができていない．

System Area Networkは，ハードウェア機能を直接用いることにより，TCP/IP相当の通信をより低オーバヘッドで実行することが可能である．そこで我々は，従来のIIOP/TCP/IPに加え，System Area Network上の低オーバヘッドなプロトコルを導入したCrisp ORBを提案する．導入したプロトコルは，下位レイヤにVIアーキテクチャ[2]，上位レイヤにSIOPを使用する．System Area Network外に対する通信は，従来通りIIOP/TCP/IPを利用することができ，選択処理はオーバヘッドなくORB内に隠蔽される．

富士通のCORBA準拠ORBであるINTERSTAGE/ObjectDirectorを元にCrisp ORBのプロトタイプを設計・実装した．VIアーキテクチャ準拠の通信ドライバとライブラリは富士通のSCnet[3]を利用した．さらに富士通のSystem Area NetworkであるSynfinity-0[4]により双方向通信の性能評価テストを行い，提案の有効性を実証した．

2 System Area Networkにおける ORB

System Area Networkは図1のようなクラスタ内ノードを高速結合したネットワークハードウェアで，TandemのServerNet[5],MyricomのMyrinet[6]，富士通のSinfinity-0(片方向bandwidth 200MB/s, 二次元メッシュ)などがある．

System Area Networkの特徴は次の3つである．

・異なるノードのメモリ領域を指定してCPUを介さずに直接データ転送が行える(Remote DMA)．

・通信距離を限定してハードウェア信頼性の高い通信が行える．

・通信処理のためのハードウェア起動をユーザレベルから直接行える．

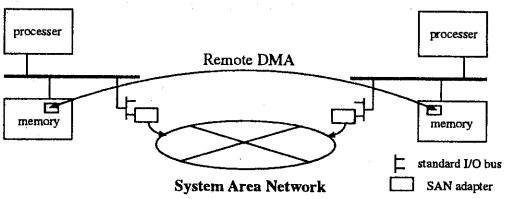

図1 System Area Network

ORBの説明とSystem Area Network上でORBを使用した場合の考察を以下に行う．

2.1 ORB

ORB上の通信では図2のように，クライアントは各サーバオブジェクトに対するリクエスト・リプライをORBに対して行う．クライアントはサーバオブジェクトの物理的な位置が透過なので，リクエストにオペレーションやパラメタしか指定しない．ORBはクライアントに変わってサーバオブジェクトの物理的な位置を解決し，サーバオブジェクトを呼び出す．

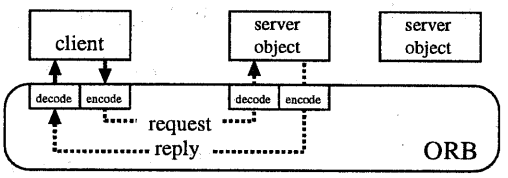

図2 ORBを用いたオブジェクト間通信

ORBはそれぞれ独自に，オブジェクトを一意に識別するオブジェクトリファレンスを持つ．標準アーキテクチャCORBAでは，ORB間の相互運用性を保つために，Interoperable Object Reference(IOR)を持つことを規定している．

またオブジェクトのプラットホームごとに，プログラミング言語の型定義やCPUアーキテクチャのbyte orderなどで，データ表現が異なることがある．透過的な通信を行うために，送信データをプラットホーム非依存なデータ表現へエンコードし受信側でデコードを行う必要がある．

このデータ表現の変換を含めた通信処理はORBで行われる．CORBAではORB間の相互運用性を保つためにGIOP/IIOPを規定している．GIOPはプラットホーム非依存なデータ表現としてCommon Data Representation(CDR)を規定している．また，

コネクション指向な通信を前提としており，様々な通信層プロトコルにマッピングすることが可能である．IIOP はTCP/IP へのマッピング規則であり，相互運用性を保つために実装することが義務づけられている．

2.2 System Area Network 上での ORB 通信オーバヘッドの分析

System Area Network のような高速通信ハードウェアでは，短いメッセージの通信においてソフトウェアによる通信オーバヘッドが性能ボトルネックとして表れやすくなる．CORBA 準拠の ORB は，GIOP と TCP/IP を合わせたものが通信オーバヘッドになるが，Schdmit らは，そのほとんどはTCP/IP であることを示した[7]．我々が富士通のCORBA 準拠ORB である INTERSTAGE/object director を元に，SUN のプロファイルツール gprof[8] を用いて，IIOP/Synfinity-0 での短いメッセージの Round Trip Time を測定した結果，表 1 のように TCP/IP 処理が 242 μ s で，全体の 44%を占めることがわかった．

表 1 TCP/IP 処理の割合

	時間	比率
TCP/IP	242us	44%
TCP/IP 以外	308us	56%
全体	550us	

しかし System Area Network では，次のようにハードウェア機能を直接利用することで，上記TCP/IP 部分に相当する通信を，より短い時間で行うことが可能である．

- System Area Network はハードウェア信頼性の高い通信を行うことができるので，エラーレートの高い LAN/WAN での通信を前提としたTCP/IP のエラー制御は必要ない．
- System Area Network はユーザバッファを通信バッファ領域として登録し直接データ転送することができるので，通信バッファ領域にコピーする必要がない．
- System Area Network はユーザレベルの通信が行えるので，データ通信時にシステムコールを発行する必要がない．

表 1 の結果と，現在開発中である VI アーキテクチャをハードウェア実装した System Area Network の短いメッセージの Round Trip Time 算定値 56us から予測すると，187 μ s が削減され，34%の改善率が見込まれる．

3 低オーバヘッド通信を導入した Crisp ORB の提案

以上の分析より，通信オーバヘッドを削減するため，従来の TCP/IP だけではなく，図 3 のような低オーバヘッドなプロトコルを System Area Network 上で導入した Crisp ORB を提案する．

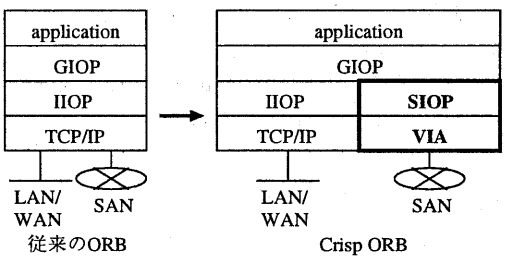

図 3 Crisp ORB のモジュール構成

GIOP から LAN/WAN を利用する場合の IIOP，TCP/IP スタックの他に，System Area Network を利用する場合の SIOP，VI アーキテクチャスタックを追加し，ネットワークに応じた処理の選択を ORB 内で自動的に行う．以下では，VI アーキテクチャ，SIOP，IIOP と SIOP の自動選択について説明を行う．

3.1 VI アーキテクチャ

VI アーキテクチャは，System Area Network 上の様々なユーザレベル通信を元に，標準化を目指して提案されたアーキテクチャで，COMPAQ,Intel,Microsoft など多くの企業がサポートしている．

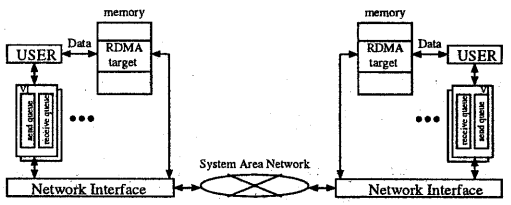

図 4 VI アーキテクチャ

VI アーキテクチャの構造を図 4 に示す．通信エンドポイントとして仮想インタフェース (VI) を作成しコネクションを開設して通信を行う．VI ごとに送信キュー・受信キューを持っており，送受信命令をキューイングして処理を非同期に行う．送受信用のユーザバッファを物理メモリ上に固定し，System Area Network の通信バッファ領域として登録するこ

とで，カーネルを介さない低オーバヘッドな通信を実現する．VI アーキテクチャを使用するための低レベル通信用 API として，Intel より VIPL[2] が提示されている．

VI アーキテクチャを採用すると次のような利点が得られる．

エラー制御処理の削減

ハードウェアの信頼性のある通信を利用して，エラー制御処理を行わない通信が可能である．

コピーの削減

ユーザ空間の送受信データ領域を，VI アーキテクチャの通信バッファ領域に登録することができるので，copyin，copyout に相当するプロトコルレイヤ間の通信バッファのデータコピーが削減できる．

システムコールの削減

メッセージ通信処理開始のためのハードウェアへの起動指示を，カーネルの関与なしにユーザレベルから行えるので，システムコールのオーバヘッドを削減できる．

3.2 SIOP

SIOP は，GIOP の要求する通信を VI アーキテクチャ上にマッピングするプロトコルで，以下のように定義する．

GIOP の前提条件を満たした通信

GIOP は下位のスタックに対して，データストリームコネクションを開設し，順序保証をしながらメッセージ転送を行うことを要求している．この条件を満たすように，VIPL 通信プリミティブを使用して GIOP メッセージの転送を行う．

CDR バッファの通信バッファ領域化

CDR のエンコード・デコード時に使用するバッファ (CDR バッファ) を，図 5 のように通信バッファ領域とし直接データ転送する．

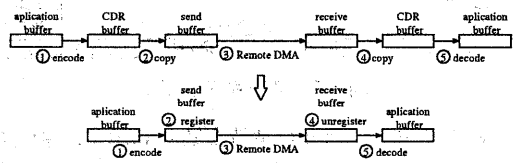

図 5 CDR バッファの通信バッファ領域化

エンドポイント作成時の一括した通信バッファ領域の登録

エンドポイント作成時に，一括して通信バッファ領域を登録してプールし，送受信時に CDR バッファとして使用する．

送受信の度に CDR バッファを通信バッファ領域として登録するのは，ページ固定のためのシステムコールが毎回伴うので効率的ではない．そこでエンドポイント作成時に，図 6 のように通信バッファ領域として一定量のメモリを一括して登録しプールしておき，必要に応じて CDR バッファとして使用するようにする．

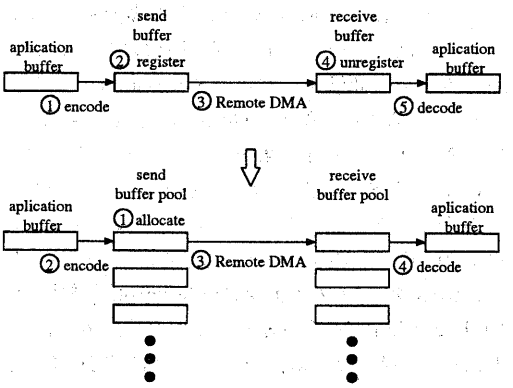

図 6 ページ固定オーバヘッドの削減

3.3 IIOP と SIOP の自動選択

ORB は，System Area Network 単独ではなく，LAN や WAN と混在した状況で使用されることが多く考えられる．System Area Network 内かどうかで，通信先の違いをユーザに意識させるのは，ユーザに大きな負担がかかる．そこでクライアントからリクエストが発生すると，サーバオブジェクトの IOR 内のホスト名をみてネットワークを判断し，ORB 内で最適な通信処理の選択を行う．ユーザは，SIOP を使う場合でも IIOP を使う場合でも IOR を透過的に使用する．

4 プロトタイプの設計と実装

提案した Crisp ORB について，富士通の CORBA 準拠 ORB である INTERSTAGE/Object Director を元にプロトタイプの設計と実装を行った．VI アーキテクチャモジュールは富士通の SCnet を使用する．SCnet は，VIPL の full conformance 機能をドライバとライブラリで実現しており，VI-NIC としてハードウェアが備えていない機能は，ドライバがエミュレーションすることで実現している．

以下に，データ通信を中心とした ORB のリクエスト・リプライ処理，フロー制御について設計・実装の説明を行う．

4.1 リクエスト・リプライ処理

今回のプロトタイプでは，System Area Network の特徴を利用した通信オーバヘッドの改善によって，最も効果が期待できる，短いメッセージの通信性能評価に重点を置いた．そこで，CDR バッファに直接通信バッファ領域を割り当てることはせず，CDR バッファと通信バッファ領域の間でコピーを行う．提案した手法に比べ長いメッセージでは性能低下するが，短いメッセージでは変わらない性能を出すことができる．全体のフローを図 7 に示す．

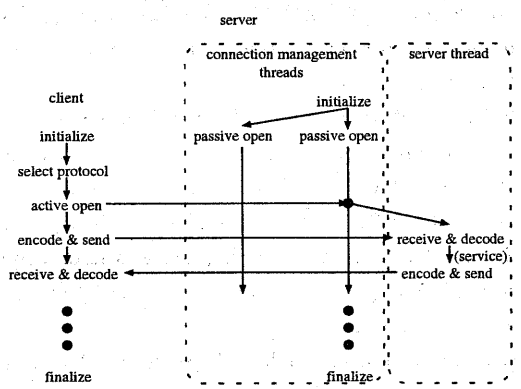

図 7 リクエスト・リプライ処理

まず各ホストで，System Area Network に対するサブネットを登録し，サーバオブジェクトを System Area Network に対するホスト名でインストールする．

プロセスが起動すると VIPL の初期化 (initialize) を行う．サーバ側は VIPL と TCP/IP の passive open をマルチスレッドで行う．リクエストが発生すると，クライアント側はサーバオブジェクトの IOR を取得し，ホスト名の IP アドレスが，登録された System Area Network のサブネット内であるどうかを判断する．サブネット内であれば VIPL で active open し提案した SIOP 通信を，そうでないならば TCP/IP で active open し従来の IIOP 通信を開始する．

このとき SIOP 通信では，エンドポイント作成時に 128KB のメモリを通信バッファ領域として登録する．登録したメモリは，送受信用各 16 個の 4096B のバッファとして用意しておく．

CDR バッファから送信バッファへデータをコピーし送信 (send) する．受信 (receive) したデータは受信バッファから CDR バッファへコピーする．

コネクション切断時，SIOP 通信では登録した通信バッファ領域を解放する．

全ての通信が終了後，プロセスが終了する前に VIPL の終了処理 (finalize) を行う．

4.2 フロー制御

VI アーキテクチャでは，フロー制御を上位レイヤで行う必要がある．まずエンドポイント作成時に登録した送受信バッファに番号を付ける．バッファは順番に循環して使用する．用意した受信バッファの数をトークンの数として送信側へ通知する．以後送信側は，トークンの数だけ送信することが出来る．ORB の性質上，双方向の通信が多いと考えられるので，通常時にはトークンを受信側が次の送信 (リプライ) を行うときにメッセージに添付して送り返す．ただし送信側でトークンが枯渇しないように，不足時には送信側からトークンを要求できるようにしている．この際デッドロックが起きないように，データ用とは別にフロー制御用のコネクションを用意する．

5 性能評価

富士通の System Area Network である Synfinity-0 を用い，双方向通信によるプロトタイプの性能評価テストを行った．以下に評価結果を示す．

5.1 評価方法

測定は Sun Ultra 30 2 台を使って行った．1CPU の UltraSPARC-II 296MHz でメモリは 128MB である．通信ハードウェアは FastEther(100Mb/s, PCI バス)，Synfinity-0(200MB/s, PCI バス) を用い，IIOP/Fast Ether，IIOP/Synfinity-0，SIOP/Synfinity-0 の 3 種類の状況下での Round Trip Time を比較測定した．テストプログラムは，string 型のリクエストに integer 型のリプライを即座に返す，単純な双方向通信 (ping pong) である．このときのリクエストメッセージ長は，IOR などを含んだ ORB ヘッダ長 (132B) と null バイトを含めた string 長の合計となり，リプライメッセージ長は，ORB ヘッダ長 (20B) と integer(4B) を合わせた 24B である．

5.2 評価結果

null バイトを入れた string 長 1, 100, 200, 300, ... , 2000B で測定を行った．図 8 に結果のグラフを示す．測定値は 10 万回繰り返したときの 1 回あたりの平均値である．

従来の IIOP/Synfinity-0 では，IIOP/FastEther とほとんど性能差はなく，逆に null string では遅くなっていた．提案した SIOP/Synfinity-0 は null string で 414 μ s,IIOP/Synfinity-0 に対する改善率は 25%という結果になった．

これは，2 章で予測した改善率である 34%よりも

小さい．現状の SCnet では，SCnet プロトコルヘッダと転送データの集約，NIC 上のデータキューの排他などが，ソフトウェアエミュレーションにより実装されているからである．現在開発中の VIPL を直接ハードウェアサポートする System Area Network を利用すれば，さらに性能改善が期待できる．

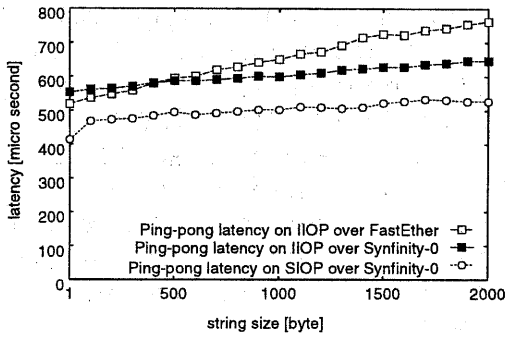

図 8 Round Trip Time 測定結果

6 関連研究

Madukkarumukumana らは，通信ミドルウェアの DCOM に対して，Myrinet 上の VI アーキテクチャのソフトウェアエミュレーションで，通信性能を改善している [9].

Schmidt らは，ATM を効率的に使用する方法について述べている [10].

7 結論

本論文では，System Area Network 上の ORB 通信オーバヘッド削減のために，従来の IIOP/TCP/IP に加え，System Area Network 上の低オーバヘッドなプロトコルを導入した Crisp ORB の提案を行った．

導入したプロトコルは，下位レイヤに VI アーキテクチャ，上位レイヤに SIOP を使用したもので，バッファ間のコピー，システムコール，エラー制御などを削減している．また，System Area Network 外に対する通信は従来の IIOP/TCP/IP で行うことができ，処理の選択はオーバヘッドなく ORB 内に隠蔽される．

提案した Crisp ORB のプロトタイプを設計・実装し，双方向通信の性能評価テストを行った．null string の Round Trip Time は 414 μ s で改善率は 25%であった．提案した手法により従来 TCP/IP 処理で占められていた 44%のオーバヘッドを半分以下に削減し性能向上することを実証した．

参考文献

[1]OMG; “CORBA/IIOP 2.2”; http://www.omg.org/corba/corbaiiop.html

[2]Intel Corp; “Intel Virtual Interface (VI) Architecture Developer’s Guide Revision 1.0”; September 9.1998

[3]Andreas SAVVA et al; “Smart Cluster Network (SCnet): A next generation communication interface for the AP-Net”; PCW ’98 September 7-8, 1998, pp. 45-51.

[4]O.Shiraki et al; “AP-NET advanced high- performance network for scalable parallel server”; In Hot Interconnects, 1996

[5]Robert W.Horst,David Garcia; “ServerNet SAN I/O Architecture”; Hot Interconnects V,August 1997 ,http://www.tandem.com/Library.asp

[6]Nanette J.Boden et al; “Myrinet–A Gigabit-per-Second Local-Area Network”; IEEE Micro, Feburary 1995,http://www.myri.com/research/publications/index.html

[7]Andy Gokhale,Douglas C. Schmidt; “Measuring and Optimizing CORBA Latency and Scalability Over High-speed Networks”; IEEE Transaction on Computing,http://www.cs.wustl.edu/ schmidt/corba-research-performance.html

[8]Sun Microsystems, Inc. gprof manual page. Solaris 2.6 manual page.

[9]Rajesh S.Madukkarumukumana et al; “Harnessing User-Level Networking Architectures for Distributed Object Computing over High-Speed Networks”; Proceedings of the 2nd USENIX Windows NT Symposium Seattle, Washington, August 3-4,1998

[10]Aniruddha Gokhale,Douglas C. Schmidt; “Principles for Optimizing CORBA Inter Inter-ORB Protcol Performance; Proceedings of the HICSS