

CORBA 準拠 ORB のリアルタイム化における 緊急要求処理への適用可能性の評価

2N-7

武本 充治 中村 隆幸 田中 聡 久保田 稔

NTT 光ネットワークシステム研究所

1 はじめに

我々は、将来の通信システムを支える抜本的なネットワークアーキテクチャに適用することを狙いとして、分散処理プラットフォーム DONA DPE[1] を検討している。これは、他のネットワーク事業者やサービス事業者との接続を考慮し、システムに高い相互接続性を持たせることを目的の1つとしている。この相互接続性を達成する1つの方法として、有力な分散処理環境の標準である OMG の CORBA[2] に基づいた ORB (Object Request Broker) (以下、DONA ORB) を実装している [3]。本稿では、通信サービスに必須であるリアルタイム性能を、CORBA 準拠 ORB 上で実現する方式について述べる。また、実装中の DONA ORB を用い、リアルタイム性能の測定を行なった結果を示す。

2 分散処理とリアルタイム性

今後、分散システムを構築する上では、標準化された ORB の採用が有力である。その理由は、(1) 分散透過性・相互接続性を利用して、柔軟なアプリケーションの構成が実現できることと、(2) 通信に関する機構をアプリケーションに対して隠蔽することにより、高いアプリケーションの記述性を達成できること、である。つまり、ORB を採用することにより、実行環境である ORB とアプリケーションの開発を分離が容易になる。

本稿で取り扱うリアルタイム性の保証とは End-to-end の実行時間の保証とする。リアルタイム性を保証するには、CPU や通信路などの、その実行に関係するすべてのリソースについて実行時間の予測が可能である必要がある。したがって、どのリソースが使用されるかが実行前には分からない分散処理環境において、リアルタイム性を保証するのは一般に困難である。

しかし、通信サービスは本質的に分散システム上に実現されるものであり、しかも、リアルタイム性を保証することが求められる。既存の通信サービス [4] では、下層の OS から上層のアプリケーションまで、一括で開発することにより、リアルタイム性を保証してきた。

以上より、今後、分散システム上で通信サービスを柔軟に開発するためには、ORB 上にリアルタイム性を保証する機構を実現する必要がある。

現在、リアルタイム性を実現する機能を持つ Real-

time CORBA1.0 の仕様を OMG において作成中である [5]。また、ORB 自体を軽量化することでリアルタイム性を保証する研究 [6] もなされている。

リアルタイム性を実現するためにはシステム全体が連携して動作する必要がある。将来の分散オブジェクト指向によるサービスでは、OS とアプリケーションの中間に位置する ORB がリアルタイム性のための機能を持つとともに、これらの間の橋渡しをすることが重要になる。

図1に示すように、ORB のリアルタイム化を実現する方法を分類すると3種類になる。これらの内、(1) RT-OS 機能を用いた実行順序制御を本稿で取り扱う。なお、[6] などと同様に、(2) ORB の処理軽量化を DONA ORB をターゲットとしても検討・実装している [3]。また、(3) アプリケーションの通信最適化支援機構については [7] で報告される。

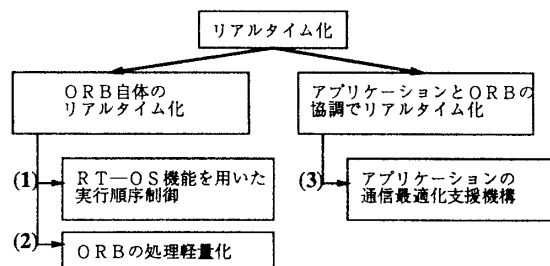


図1: ORB リアルタイム化の取り組みの分類

(1) は、RT-OS の優先度にしたがったスケジューリングで実現されるものである。Realtime CORBA1.0 においても、リクエストの優先度の取り扱いは中心的な機能として考えられている。しかし、一般に RT-OS 上にタスクとして生成されるソフトウェア機能が大きいと、スケジューリングの効果が弱まる。この大きなソフトウェア機能の例として、ORB に含まれる通信プロトコルを操作する機能や、ORB 上のオブジェクトのオペレーション呼出における複雑な引数を解析する機能などが挙げられる。

つまり、(1) について、ORB 上にオブジェクトを実装する形態で RT-OS の機能がどの程度効果があるかが明白になっておらず、それを評価する必要がある。

3 ORB におけるリアルタイム性能の測定

3.1 測定の環境と方法

本稿で報告する性能は、クライアントオブジェクト (以下、クライアント) から、遠隔にある ORB 上のサーバオブジェクト (以下、サーバ) にリクエストを送信し、そのリクエストがサーバで処理され、結果がクラ

Evaluation of the realtime facility to process the urgent requests on a CORBA-compliant ORB

Michiharu Takemoto, Takayuki Nakamura, Satoshi Tanaka, and Minoru Kubota

NTT Optical Network Systems Laboratories

3-9-11 Midori-cho, Musashino-city, Tokyo, 180-8585, Japan

takemoto@ma.onlab.ntt.co.jp

クライアントに戻ってくるまでの round-trip time を測定したものである。これを処理時間と呼ぶ。これらのオブジェクトはタスクあるいはプロセスで実現される。

本測定の目的は、サーバのあるホストにおいて、通常のレベルのリクエストの処理中に、緊急のレベルのリクエストを適切に処理できるかを確認することである。そのため、図2に示すように、測定するクライアントとは別にサーバへリクエストを出す負荷生成クライアントを用意した。

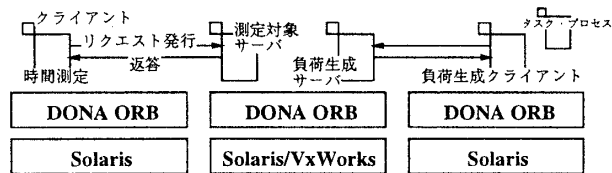


図2: 測定環境

リクエストのレベルの相違は、リクエストを処理するサーバを実現するタスク・プロセスに与える優先度の相違で実現している。また、負荷を変化させるために、負荷生成クライアントが呼び出す負荷生成サーバのオペレーションは、引数を変化させることで、オペレーションの処理時間が異なるようにした。

以上のプログラムを用い、負荷として測定処理用のサーバと(1)同じ優先度の負荷生成サーバを使用した場合と、(2)低い優先度の負荷生成サーバを使用した場合について、処理時間を測定した。さらに、負荷生成サーバが存在しない場合についても処理時間を測定した。実行環境は、RT-OSとしてVxWorks (Pentium 133MHz)と非RT-OSのUNIXとしてSolaris2.5.1 (UltraSPARC 167MHz)を用いた。

3.2 測定結果

図3に上述の測定時間を度数分布で示す。横軸が1つのオペレーションにかかる時間をクライアント側でround-tripとして計った時間(単位マイクロ秒)で、縦軸がオペレーションの個数である。1clientは負荷がない場合を、high_prioは優先度の低い負荷がある場合を、same_prioは同じ優先度の負荷がある場合を、それぞれ示す。なお、RT-OSの方が全体的に遅いのは、CPU性能の違いによる。

3.3 リアルタイム性の評価

RT-OSでは、優先度の高い処理については、負荷の有無に関わらず、処理時間の分布のピークはあまり変化しておらず、裾もあまり広がらない。つまり、高負荷時にも緊急の要求レベルのリクエストが迅速に処理でき、高いリアルタイム性を達成できる。通常RT-OSで動作することを期待される小さなタスクだけでなく、ORB上のオブジェクトの実行のような通信プロトコル処理が関係する大きなタスクである場合でも、高いリアルタイム性を達成している。

これに対して、非RT-OSのUNIXの場合、負荷がない場合は、ピークが鋭く左側に出るものの、負荷がある場合は、ピークがなだらかになり、処理時間が増加する。負荷の優先度を下げ、相対的に目的とするサーバの

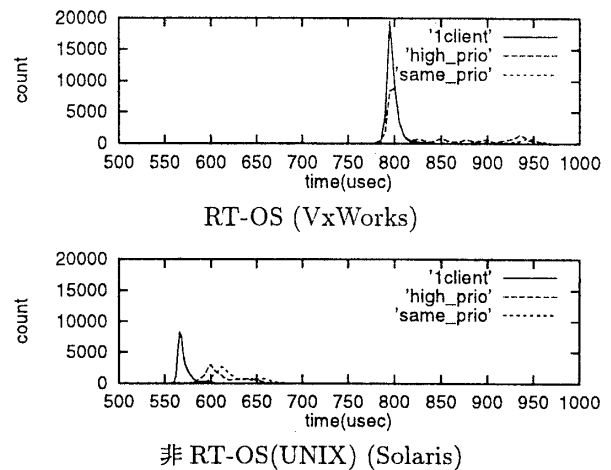


図3: 測定結果

優先度を上げても、ピークが負荷のない時のように鋭くならない。つまり、負荷がある時に緊急の要求レベルのリクエストの実行を行なった場合、リアルタイム性の確保は難しいといえる。

また、今回の実験が多重度2であることを考え合わせると、多重度がさらに上がる実システムの場合、ORBに優先度を守らせる各種機能を組み込んでいく必要があるといえる。

4 おわりに

本稿では、ORB上のオブジェクトにリアルタイム性を保証するための機構の実現方法の分類について述べ、RT-OSの機能である優先度を適切に使用することで、リアルタイム性を保証することができることを実測により示した。

今後は、サーバ側のオブジェクトを増やすことでの多重度がさらに向上した場合の性能特性や、複雑な構造を持つ引数のオペレーションを呼び出すことでの引数の処理に時間がかかる場合の性能特性の分析を行ない、ORB上のリアルタイムシステム構築に必要な機能の詳細化を行なう予定である。

参考文献

- [1] Suzuki, S. et al., "DONA: Distributed Object-Oriented Network Architecture for Revolutionary Network Reconstruction," in *ISS*, vol. II, pp. 459-465, 1997.
- [2] Object Management Group, "The Common Object Request Broker: Architecture and Specification (Revision 2.2)," Feb. 1998. <http://www.omg.org/>.
- [3] 中村 隆幸, 他, "分散オブジェクト実行環境の軽量化," 情処全大 3F-03, Oct. 1998.
- [4] 鈴木 滋彦, "新ノードシステムの開発," *NTT R & D*, pp. 497-506, June 1996.
- [5] Object Management Group, "Realtime CORBA 1.0 Request for Proposal," Sept. 1997. <http://www.omg.org/pub/docs/orbos/97-09-31>.
- [6] Schmidt, D. C. et al., "A High-Performance End System Architecture for Real-Time CORBA," *IEEE Commun. Mag.*, vol. 35, no. 2, pp. 72-77, Feb. 1997.
- [7] 中村 隆幸, 他, "CORBA準拠ORBのリアルタイム化におけるアプリケーション記述支援機構," 情処全大 2N-05, Mar. 1999.