

ネットワーク管理システム実装への CORBA の適用法に関する実験的考察

2U-5

服部 元 吉原 貴仁 杉山 敬三 小花 貞夫
(株)KDD 研究所

1. はじめに

近年、分散オブジェクトプラットフォームとして OMG(Object Management Group)が規定する CORBA (Common Object Request Broker Architecture)を用いた各種システム構築が進んでいる。ネットワーク管理の分野においても、CORBA を適用した網管理システムの構築事例[1-4] や SNMP(Simple Network Management Protocol)や CMIP (Common Management Information Protocol)を具備する装置をCORBA環境に收容するシステムの構築や実装の報告[5] などがある。

CORBA によるネットワーク管理システムの構築では、アプリケーションの実装方法は様々であり、選択すべき CORBA パラメタも多数存在するが、それらを決定する指針は示されていない。本稿では、ネットワーク管理システムにおける通信に対し、基本的な通信処理性能やパラメタ選択による影響を測定し、アプリケーション実装方法や、CORBA パラメタ選択に関する指針を示す。

2. CORBA によるネットワーク管理システム構築

ネットワーク管理の通信形態は一般に、管理操作と通知に大別できる。管理操作ではマネージャ(クライアント)からの要求に対しエージェント(サーバ)から応答を返す。クライアントから要求発行時には、サーバオブジェクトの識別子であるオブジェクト・リファレンスの取得やオブジェクトの生成等の操作が行われる。また、現在広く普及する SNMP 装置の收容や、並行処理を行うスレッドのサーバへの実装の必要がある。通知では、同一の情報を単一の宛先だけでなく複数の宛先に同報する形態が存在する。これらを CORBA を用いて実現し、性能試験を行う。

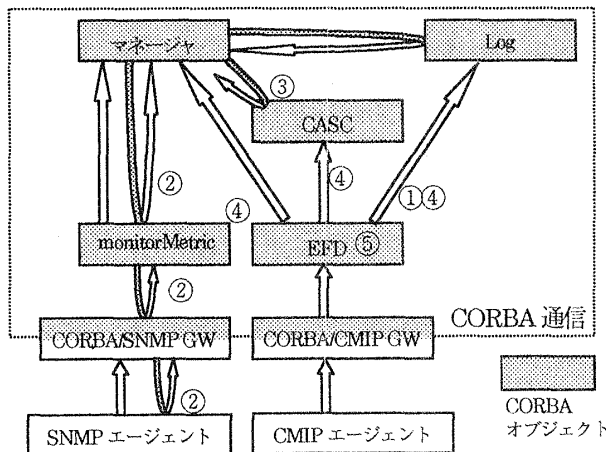


図1: CORBA による NW 管理システムのモデルと試験構成

3. 本試験の構成と試験項目

試験における構成を図1に示すとともに、以下に本試験で行う試験項目について述べる。

3.1 試験構成

図1において、EFD(Event Forwarding Discriminator), Log, CASC (Current Alarm Summary Control), monitorMetric は各々、ITU-T 勧告 X.734, X.735, Q.821, X.739 で規定される管理オブジェクト相当の機能を果たす。また、既存の SNMP 及び CMIP エージェントは、先に開発した CORBA/SNMP ゲートウェイ(SNMP G/W)及び CORBA/CMIP ゲートウェイ (CMIP G/W)を用いて收容し、CORBA エージェントとして用いる。これにより、図1のマネージャと monitorMetric 間でポーリングを、EFDとマネージャ/CASC/Log 間で通知の同報を実現する。

3.2 試験項目

CORBA製品には固有のパラメタが存在する場合があるが、製品に特化した試験を避けるため、CORBA標準のパラメタのみを使用する。以下に、今回の検証で使用した試験項目及び試験方法を、基本的な通信とパラメタ選択項目に分けて示す。

(A)基本的な通信処理性能

① オブジェクトの生成時間

オブジェクトを多数生成するアプリケーションへの影響を測定する。具体的にはEFDからの通知を基にLogでアラームレコードオブジェクトを生成する際の処理時間を測定する。

② SNMPエージェントからの要求応答時間

基本的な管理操作要求に対するCORBAのオーバーヘッドを測定する。具体的にはmonitorMetricからSNMP G/WへGET要求を行い、その応答時間を測定する。

(B)パラメタ選択項目による影響

③ オブジェクト・リファレンス取得

オブジェクト・リファレンスの取得方法として、
a1) CORBA製品が提供するバインド関数を使用し、かつホスト名を指定する場合
a2)バインド関数を使用しホスト名を省略してホストの検索を自動的に行うローケータを利用する場合
b) オブジェクトの名前から検索するサービスであるNS (Naming Service)を利用する場合
の3通りが存在する。これらの項目に対し、マネージャがCASCのオブジェクト・リファレンス取得に要する時間を比較する。

④ イベント通信

イベント通信方式として、
a) 応答のない一方通信であるOneway通信
b) マルチキャスト通信を実現する共通オブジェクトサービスのイベントサービスを利用した通信
の2通りが実現できる。これらの項目に対し、EFDからマネージャ、CASC、Logへのイベント配信時における、EFD内のメッセージ処理時間を比較する。

⑤ スレッドの割当て

複数のオブジェクトへの送信要請に対する処理を、

- a) スレッドを利用しない場合
 - b) スレッドを利用する場合
- の2通りで行う。これらの項目に対し、EFDからマネージャ、CASC、Logへのイベント配信の際に、EFD内のメッセージ処理時間を比較する。

4 試験結果

図1の構成に基づき、3. 2の各項目の時間を測定した。ここではCORBA製品としてIONA社のOrbixMT2.3c(以降、Orbix)、イベントサービスとしてマルチキャストを実現するOrbixTalkを使用した。また、試験には3台のUNIXワークステーション(ULTRA SPARC:全て1CPU)を用いた。表1に各項目でのオブジェクト配置を示す。

表1:各試験項目でのオブジェクト配置

	host1(133MHz)	host2(300MHz)	host3(110MHz)
①		EFD,Log*,NS	G/W
②	CASC	Mng*,Mmc,G/W	NS
③	EFD,Mng*,Log	G/W,CASC	Mmc,NS
④	Mng1-5,EFD*	G/W,CASC1-5	Log1-5,NS
⑤-1	EFD*,Mng,Log	CASC,NS	G/W
⑤-2	CASC,NS	EFD*,Mng,Log	G/W

注1:Mng(マネージャ), G/W(SNMP/G/W および CMIPG/W), Mmc(monitorMetric)
Mng1-5(5つのマネージャMng1,Mng2,Mng3,Mng4,Mng5)。Log, CASCも同様
注2:*は測定対象のオブジェクトを示す。

①オブジェクトの生成時間

1つのアラームレコードを1つのオブジェクトとして生成し、その生成時間は平均 29.51msec となった。大規模なネットワークにおいて、障害時に 1 千件のアラームレコードを生成する必要が生じた場合、約 30 秒要することとなる。処理時間の短縮が必要な場合には、すべてのアラームレコードを単一のオブジェクトとして生成するか、またはリストの状態メモリ上に保存する等の工夫が必要である。

②SNMP エージェントからの応答時間

表 2:平均応答時間(msec)

取得MIB	monitorMetric・SNMP エージェント間(往復)	SNMP G/W の処理を除いた時間
ifInUcastPkts(GET)	21.55	15.35

SNMP G/W 内での処理時間は約 6.2msec[5]であり、これを除いた時間は 15.35(msec)となる。SNMP の GET は CORBA を介さない場合で 2~4msec であるため、monitorMetric・SNMP エージェント間では、12~13msec がオーバーヘッドとなっている。また、この G/W を用いた場合、21.55msec 以下の間隔でポーリングはできないことがわかる。しかしながら、このような短い間隔でポーリングすることはほとんどないため、実用上問題はない。

③オブジェクト・リファレンス取得

表 3:オブジェクト・リファレンス取得の平均所要時間(msec)

a1)バインド関数(ホスト名指定あり)	1.38
a2)バインド関数(ホスト名指定なし)	13.8
b)NSを用いる	6.74

1つのオブジェクトに対し管理操作を繰り返し行う場合は表3の値は無視できるが、ログレコード取得のように多数のオブジェクトを扱うアプリケーションでは、オブジェクト・リファレンス取得時間の占める割合が大きく、a1)の方法が適して

いる。ただし、a1)ではオブジェクトの位置透過性が失われるため、それが必要な場合にはNSの利用が望ましい。

④イベント通信

図2でOneway通信は、転送先のオブジェクトが3つ増加する毎に約100msecずつ増加している。イベントサービスではマルチキャスト通信を行っているため、EFD内でのメッセージ処理時間は一定となるが、OrbixTalkはメッセージ不着の際に再送する機能を有するため、右上がりとなる。

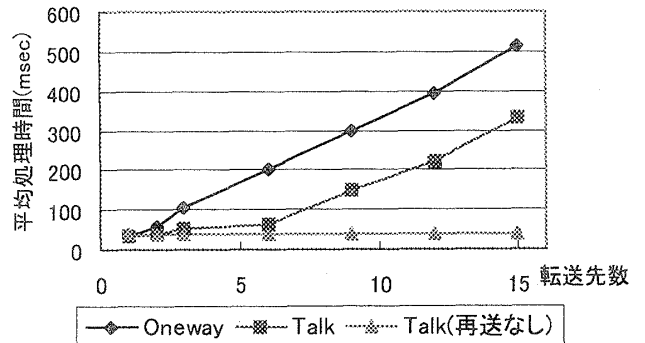


図2:EFDにおける1メッセージ処理時間

1つのマシンに複数のオブジェクトを実装した場合、受信側では未処理メッセージを保持しておくためのバッファが溢れやすくなり、再送回数が増大する。したがって、イベント送出間隔の調整など、これを意識したアプリケーションの実装が必要である。

⑤スレッドの割当て

表 4:EFDにおける1メッセージ平均処理時間(msec)

	host1(⑤-1)	host2(⑤-2)
a) スレッド未使用	274.26	137.58
b) 3スレッド使用	273.36	129.79
減少率	0.33%	5.67%

マルチスレッドを利用すると多少は速くなるが、殆ど差はない。本試験では、有効性は認められなかった。

5 おわりに

本稿では、CORBAを用いたネットワーク管理システムにおいて、メッセージ授受時間等のCORBAの基本的な通信処理性能を知る上で必要な項目やCORBAパラメタ選択による影響を測定し、アプリケーション実装時の指針を示した。他のCORBA製品毎に本稿と同様の項目の試験を行うことで、適切なCORBAパラメタの選択や性能要件に則した製品の決定を行うことが望ましい。最後に日頃ご指導頂く(株)KDD 研究所村谷拓郎所長、および鈴木健二副所長に感謝します。

6 参考文献

[1]:M.Tomono, "An Event Notification Framework based on Java and CORBA", In Proc. of *IM'99*, pp.563-576, 1999.
 [2]:R.B.Whitner, "Designing Scalable Applicationd Using CORBA", In Proc. of *IM'97*, pp.503-514, 1997.
 [3]:寺島,今井,金枝上,小林. テレコムシステムに対するCORBA適用の検討. 1999年情報処理学会, pp.73-78, Jan., 1999.
 [4]:武本,中村,片山,田中,久保田. Real-time OS上でのCORBA準拠ORBの実装と性能評価. 情報処理学会DICOMO'99シンポジウム, pp.531-536, Jun., 1999.
 [5]:堀内,吉原,小花,鈴木. 分散オブジェクトを用いたネットワーク管理におけるTMN/SNMP収容方式. 情報処理学会論文誌, Vol.40, No.1, pp.103-112, Jun., 1999.