

Win32에서 CORBA Naming Service Evaluation

이종기, *최환수

명지대학교 공과대학 전기공학과

전화 : 031-330-6363 / 핸드폰 : 019-393-4268

CORBA Naming Service Evaluation in win32

Jong Ki Lee, Hwan Soo Choi

Dept. Electrical Engineering, Myongji University

E-mail: jongki@hotmail.com

Abstract

The Naming Service is central to most common object request broker architecture(CORBA) applications. It serves as a directory for CORBA objects—the building blocks of CORBA applications. As the Windows computing environment is disseminated, the performance features of the existing CORBA is becoming of great interest. Thus, in this paper, we intend to examine some performance features of three commercially-available Naming Service implementations of CORBA in strictly Windows (NT and 98) environment.

분산된 객체를 쉽게 찾을 수 있도록 한다. 현재 CORBA standard specification을 구현한 많은 Object Request Broker(ORB)들이 있다. 그 중에서 본 논문에서는 개개의 ORB가 CORBA Naming Service를 구현함에 있어 어떤 성능 차이를 보이는지 알아보는 것이 목적이다. 특히 Windows NT 또는 Windows 98 전산환경이 많이 도입됨에 따라 이들 환경에서 상업적으로 구현된 CORBA specification의 성능에 대한 일반인의 관심이 많이 부각되었다. 따라서 본 연구에서는 CORBA ORB의 Naming Service에 대한 평가를 시도하였다.

I. 서론

Common Object Request Broker Architecture(CORBA)는 소프트웨어 객체가 분산 환경에서 작동하는 방식에 대한 일단의 명세서이다. 이 명세서는 많은 소프트웨어 업체들에 의해 결성된 Object Management Group(OMG)에서 관리하고 있다. OMG의 CORBA Naming Service는 CORBA 표준 서비스에서 가장 기본적인 서비스이다. 이 Naming Service는 분산된 객체에서 객체 참조를 쉽게 얻어올 수 있도록 한곳에 모아 이용 가능하게 해주는 것으로, CORBA application에서 클라이언트는 객체의 이름으로

II. Naming Service

OMG Naming Service는 standard CORBA specification의 가장 기본이 되는 서비스이다. Naming Service는 object에 고유한 이름을 제공하는 것이다. Object에 이름을 매핑하기 위해서는 먼저 application server(servant)에서 context에 이름을 바인드(bind)한다. 고유한 이름으로 binding된 object는 Naming Service에 등록이 된다. Client는 Naming Service에 object reference를 얻기 위해 이름을 질의를 하고 해석하여 이름에 binding된 object reference를 얻을 수 있다. 다음 그림1은 name binding과 resolution 과정을 보여주는 그림

이다.

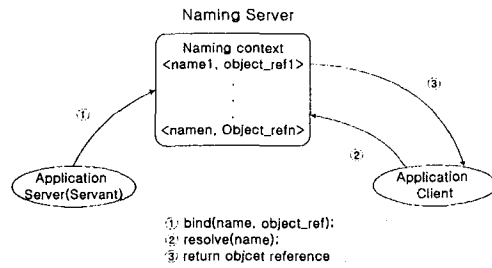


Figure 1. Name binding and resolving

그림 2에서 보여주는 것과 같이 각각의 이름은 특정 context에 바인드 되는 구조적인 형태를 갖게 된다. 이러한 구조적 형태를 naming graph라고 한다. 속이 빈 노드는 naming context이고 검은색 노드는 name과 object reference와 바인드된 것을 말한다. 이 naming graph에서 알 수 있는 것은 client에 서비스를 제공하기 위해서는 맨 마지막 검은색 노드에 이름이 바인드 되어야 한다.

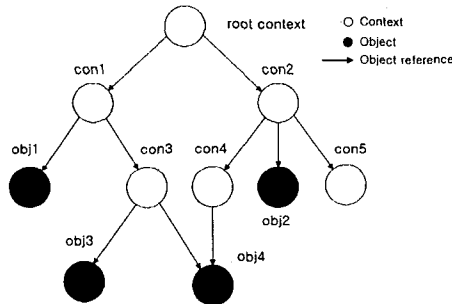


Figure 2. Naming graph

III. Evaluation Environment

본 논문에서 사용된 테스트 컴퓨터는 다음과 같다.

표 1. Computer configurations

Machine	Processor	OS	Memory	Network Adapter
Name Server	P-III 550MHz	NT 4.0	192MB	10Base-T
Client1	P-II 450MHz	NT 4.0	128MB	10Base-T
Client2	P-II 330MHz	NT 4.0	64MB	10Base-T
Client3	P-III 450MHz(dual)	NT 4.0	192MB	10Base-T
Client4	P-III 500MHz(dual)	NT 4.0	256MB	100Base-T

다음은 Naming Service evaluation에 사용될 ORB product이다.

표 2. ORB products

Vendor	Broker	Version(C++)
IONA	Orbix	3.0.1
OCI	ACE ORB(TAO)	5.0/1.0
Borland	Visibroker	3.3.3

다음은 테스트 컴퓨터가 속한 네트워크 구성 그래프이다.

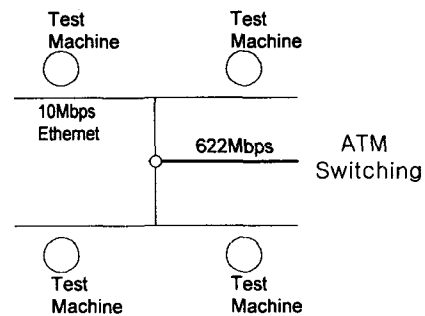


Figure 3. Network configuration

IV. Evaluation Criteria

각각의 ORB product를 테스트 위해 본 논문에서는 가장 핵심적인 3가지 테스트 영역을 규정하였다. 첫째, Standard functionality, 둘째, Scalability, 셋째, Interoperability이다. 테스트를 위해 사용될 IDL(Interface Definition Language)은 다음과 같다.

```
//test.idl
interface Calculator
{
    long Adder(in long num);
};
```

4-1 Standard functionality

이 영역은 OMG에서 제시하는 Naming Service에서 각각의 ORB가 제대로 구현하고 있는지를 분석하는 conformance와 Naming Service에서 가장 핵심이 되는 성능(performance) 테스트로 context creation, name binding, name resolution에 대해

테스트를 수행했다. 먼저 conformance 테스트는 CORBA specification 상에서 각 ORB vendor의 Naming Service 구현에 대해 알아보는 테스트이다. 성능 테스트에서 context creation은 naming context를 생성하는 new_context()를 사용하여 각 ORB의 naming server의 name space를 구성하는데 걸리는 시간을 측정하였다. 그림 4의 왼쪽은 최초 context에 연결된 context 구성 그래프이다.

Name binding은 bind()를 사용하여 최초의 context에 이름을 바인드 하는 과정으로 name binding 수를 변화시키면서 걸리는 시간을 측정하였다. 그림 4의 오른쪽은 name binding 구성 그래프이다.

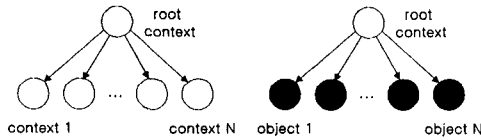


Figure 3. Naming context creation and name binding structure

Name resolution은 resolve()를 사용하여 name server에 등록된 이름을 해석하는 과정으로 bind의 수의 변화에 따른 시간 변화를 측정하였다.

4-2 Scalability

원격지 클라이언트의 수의 증가에 따른 name resolution 지연 시간을 측정하는 과정이다. name server에 2000개의 name을 바인드 하고 클라이언트에서 2000 개를 resolution 하는 시간을 측정하였다. 이 테스트는 클라이언트 수가 얼마나 name resolution 성능에 영향을 미치는지 결과를 통해서 알 수 있는 테스트이다.

4-3 Interoperability

이 테스트 영역은 각각의 ORB name server와 다른 ORB client 사이의 상호 운용 성을 테스트하는 과정으로 하나의 클라이언트에서 다른 ORB의 name server로부터 name resolution이 성공할 수 있는 지로 결정하였다.

V. Results

5-1 Standard functionality

먼저 conformance 테스트를 위해 이미 바운딩된 name server에 bind()를 사용하여 name을 다시 바운드 했을 때의 결과를 알아보았다. 모든 ORB가 제대로 이미 바운딩된 name이라고 결과를 출력했다. 또한 rebind()를 사용하여 같은 name을 바운드 했을 때 모든 ORB들이 제대로 작동했다. Name server에 등록되지 않은 name에 대해 resolve()를 했을 때도 context를 찾을 수 없었다. 그러나 Orbix의 경우에는 name을 10000 개 이상 바운드 할 때 name server의 name space에 메모리 할당 에러(OutOfMemoryError)가 발생했다.(OMG의 specification에는 이러한 name binding 개수에 대한 규정은 없다.) 다음은 성능 테스트이다.

1. Context creation

다음 그래프는 context creation 테스트 결과이다. 전반적으로 TAO가 가장 우수한 성능을 보였다. Orbix는 context creation 개수가 많아질수록 creation latency가 급격히 증가하는 것을 볼 수 있다.

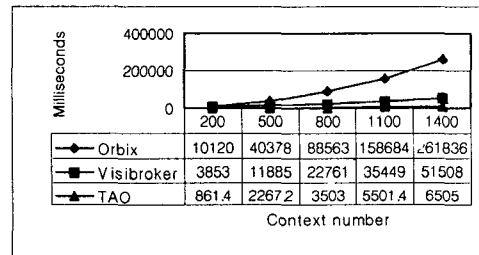


Figure 4. Context creation

2. Name binding

그림에서와 같이 orbix는 binding number가 많아질수록 지연시간이 길어짐을 알 수 있었다.

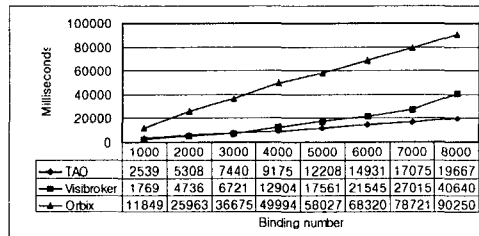


Figure 5. Name binding

3. Resolution

Visibroker, TAO는 구간에 걸쳐 time latency가 늘어나는 양이 비교적 작은 반면 Orbix는 그래프 형태를 보면 알 수 있듯이 전구간에 걸쳐 resolution 시간이 일정치 않았다.

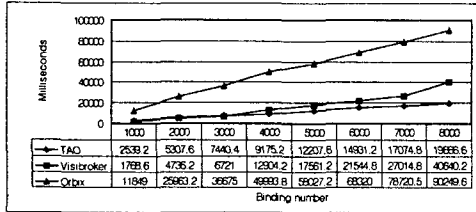


Figure 6. Resolution

5-2 Scability

Name server와 application server가 동일 컴퓨터에서 실행되는 상황에서 원격지 client를 증가 시키면서 동시에 실행시켜 테스트를 했다. 총 5대의 컴퓨터가 사용되었는데 이중 특정 컴퓨터의 데이터만 가지고 client의 resolution latency를 측정했다. Orbix는 다른 ORB에 비해 latency가 컸다.

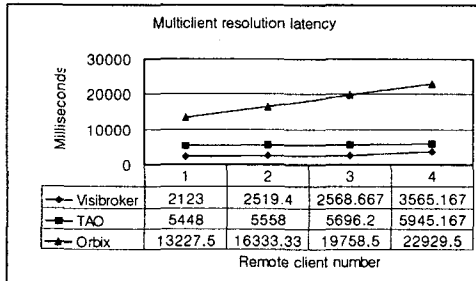


Figure 7. Multiclient resolution latency

5.-3 Interoperability

OMG의 Internet Inter-ORB Protocol(IIOP) specification에 의하면 application client는 Naming Service를 통해 application server의 object reference를 얻을 수 있다고 정의하고 있다. 하지만 OMG에서는 다른 ORB의 Naming Service를 이용하는 데에 대한 표준화된 문서를 찾아 볼 수 가 없었다. 본 논문에서는 Naming Service의 Interoperable Object Reference(IOR)를

command line의 argument로 이용하여 interoperability를 테스트 하였다. 다음과 같은 과정으로 테스트를 수행하였다.

1. 외부 Naming Service 실행
2. Application server 실행
3. Application client 실행

표3. Interoperability

ORB	Visibroker	Orbix	TAO
Visibroker	-	X	O
Orbix	X	-	X
TAO	O	X	-

위 모든 결과를 보면 Orbix의 성능이 상대적으로 저조했다. 하지만 이것은 CORBA 서비스 중에서 특정 부분에 대한 테스트이므로 어느 ORB의 성능이 우수하다고 단언할 수는 없다. 다만 Naming Service에 대한 참고는 될 수 있을 것이다. 또한 본 논문을 통해 알 수 있었던 것은 OMG의 specification이 불충분하다는 사실을 알 수 있었다. Interoperability나 Orbix의 name binding 개수 문제에서와 같이 vendor마다 OMG specification의 해석이 약간씩 다르다는 것을 알 수 있었다.

Reference

- [1] Michi Henning and Steve Vinoski, Advanced CORBA programming with C++, Addison Wesley, January, 1999
- [2] Object Management Group, Naming Service specification, December 9, 1998.
- [3] Inprise, Naming service program guide for visibroker C++ 3.3, Inprise Co. November 10, 1998
- [4] Object Computing Inc, TAO Developer's Guide, 1999
- [5] IONA, Orbix Names Programmer's Guide, IONA, 1999