

CORBA와 XML의 연동을 위한 응용 웹 서버 구조 성능 분석

이호섭^U 홍충선

경희대학교 전자정보학부

hslee@networking.kyunghee.ac.kr, cshong@khu.ac.kr

An Application Web Server Architecture Performance Analysis for CORBA and XML Integration

Ho Seob Lee^U Choong Seon Hong

School. of Electronics and Information, Kyung Hee University

요 약

컴포넌트 소프트웨어의 주된 목적은 소프트웨어 개발에 있어서 재사용을 증가시키고 개발자간의 협동과 협조를 증가시키기 위한 것이다. 이러한 컴포넌트 기술이 가져야 할 가장 중요한 특징으로는 상호운용성이 있다.

이러한 상호운용성을 위해 분산 컴포넌트의 다양한 서비스들을 웹 환경의 사용자들이 사용할 수 있게 하기 위해서, 웹 기반에서의 XML을 이용하였고, 현재 웹 기반의 환경에서 여러 가지 웹 기술들이 사용되고 있으므로, 이들 기술들을 사용하여 비교할 만한 가치가 있다. 따라서 본 논문에서는 분산 객체 패러다임중 하나인 CORBA의 컴포넌트들과 웹 환경의 사용자들의 상호운용성을 위해 CORBA와 XML의 연동을 위한 응용 웹 서버의 구조를 제안하고, 응용 웹 서버 구조의 성능 분석을 한다.

I. 서 론

인터넷은 현재 우리의 생활에 가장 많은 영향을 미치고 있는 분야 중 하나이며, 많은 프로그램이 웹을 중심으로 한 인터넷 프로그램을 구축되고 있고, 기업들도 인터넷을 활용한 정보공유와 경제활동에 초점을 맞추고 있다. 이와 같이 인터넷을 이용한 정보 제공은 사용자가 다른 어플리케이션을 설치하지 않아도 웹 브라우저만으로 서비스를 이용할 수 있고 플랫폼에 상관없이 정보를 이용할 수 있기 때문에 다른 어떤 매체보다 우월하다. 또, 분산 환경 하에서 어플리케이션간 통신 수단으로써 웹을 이용하고자 하는 흐름은 XML(eXtensible Markup Language)[1]이 출현한 이후에 보다 가속화되고 있다.

현재 가장 널리 사용되고 있는 분산 컴포넌트 기술은 OMG(Object Management Group)의 CORBA(Common Object Request Broker Architecture)[2]와 Microsoft의 COM+, Java Beans등이 존재한다. CORBA와 XML의 상호연동을 위해서 DOM(Document Object Model)[3,4] API를 사용하여서 컴포넌트 기반 기술과 웹과의 호환성을 해결하였다. 또한 이러한 컴포넌트 기반 기술과 웹과의 호환성 문제를 해결하기 위해서 XML을 이용하고자 하는 많은 노력이 있어왔다. XML은 플랫폼 및 프로그래밍 언어에 독립적이고, 확장 가능하며, 그 자체

만으로도 데이터와 메시지 전달을 하는데 있어 표현 수단으로 사용될 수 있다. 또한 DOM과 같은 표준 API를 통한 접근 방법을 제공하고, XML 스키마를 통해서 데이터 타입을 확장시킬 수 있으며, 특히 HTTP 프로토콜을 이용함으로써 웹 환경에 곧바로 이식할 수 있는 특징을 가진다.

본 논문에서는 CORBA 플랫폼이 아닌 사용자가 웹을 통해서 CORBA 플랫폼의 서비스를 이용할 수 있는 시스템을 제안하고 [2], 제안한 응용 웹 서버에 사용될 수 있는 기술들과 비교[1], 분석하고자 한다.

본고 2장에서는 본고에서 제안하는 CORBA와 XML연동을 위한 Application Web Server 구조를 제안[2]하고, 3장에서는 본 제안 구조와 다른 구조들을 사용한 응용 웹 서버의 시뮬레이션 결과를 기술하고, 4장에서는 결론에 대하여 기술한다.

II. 제안구조

이제 본 논문에서 제안하는 CORBA와 XML의 연동을 위한 Application Web Server구조를 기술할 것이다.[1,2]

본 구조에서는 Servlet을 사용하였다. Servlet를 사용함으로써 Web서버는 쓰레드 단위로 처리를 하기 때문에 초기화에 따른 오버헤드가 없고, 클라이언트가 증가하더라도 시스템의 성능의 감소현상이 없고, 기존 Java API를 모두 사용할수 가 있고, 자바 Applet과 연동이 가능하고, HTTP의 Connectionless특성에 의한 네트워크 오버헤드를 줄일수 가 있기 때문이다.

CORBA와 XML연동구조의 구성은 Web 클라이언트, Application Web 서버, CORBA 객체, Database 네 가지 요소로 구성되 있다. 이 것중 Application Web 서버는 크게 5가지 Manager로 나뉘어 진다. Web Manager, XML Document Manager, SCM(Schema Creation Manager), CORBA Client, XML Parsing Manager이다. XML Parsing Manager는 XML Tree Builder와 XML Translator으로 구성되 있다.

그림 1은 CORBA와 XML의 연동구조를 나타낸 것이다.

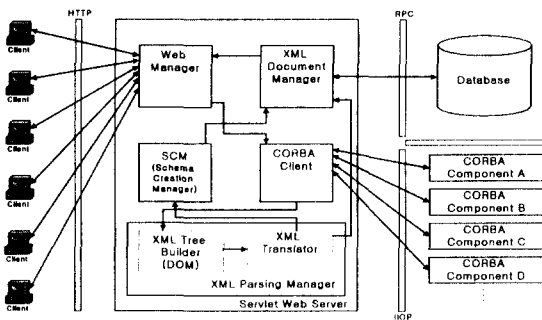


그림 1 분산환경에서 Web기반의 CORBA와 XML의 연동 구조

- Web Manager : 클라이언트에게 요청/응답을 하고 CORBA Client에게 요청을 한다.
- XML Document Manager : XML 문서들을 Database에 저장하고 관리한다.
- SCM(Schema Creation Manager) : DTD와 XML문서로 XML Schema 생성한다.
- CORBA Client : Web Manager의 요청을 받아서 CORBA 서버 객체에 요청
- XML Tree Builder : IDL을 XML DTD로 변환하고, CORBA 객체의 결과 값을 DOM Tree형태로 만든다.
- XML Translator : XML 문서를 생성한다.

이 구조를 구체적으로 살펴보면, Web 클라이언트가 Servlet 기반의 웹서버에게 요청을 하면, 이 요청은 Servlet기반 웹 서버의 Web Manager가 요청을 받는다. Web Manager는 단순히 Web 문서의 Management의 기능을 한다. 이 요청은 CORBA 클라이언트에서 전송을 한다. 그러면, 이 CORBA 클라이언트에는 등록된 CORBA 서버객체의 정보와 IDL을 가지고 있다가 Web Manager의 요청이 오면 등록된 서버객체에게 Web 클라이언트가 요청한 사항을 요청을 한다. CORBA 서버 객체는 CORBA의 요청을 처리한 다음 다시, CORBA 클라이언트로 결

과 값을 반환한다.

CORBA 클라이언트는 CORBA 서버 객체에게서 받은 결과 값과 IDL을 XML Parsing Manager의 XML Tree Builder에게 전송을 한다. 여기서 IDL을 함께 전송하는 이유는 이 IDL을 XML DTD로 변형을 위한 것이다.

XML Tree Builder에는 XML 문서를 위한 응용프로그램 인터페이스인 DOM(Document Object Model) API 함수를 이용하였다. XML 문서를 위한 응용프로그램 인터페이스에는 DOM과 SAX(Simple API for XML)가 있다. DOM은 문서를 접근하고 조작하기 위한 방법으로서 문서의 논리적 구조를 정의하고 있어 사용자들은 문서를 생성하고, 그 문서의 구조에 따라 검색하고, 엘리먼트와 문서 내용을 추가, 수정, 삭제할 수 있다. SAX는 이벤트 시퀀스로 XML 문서의 정보가 접근하는 이벤트 기반의 XML분석을 위한 표준 인터페이스이다. SAX는 핸들러를 SAX 파서와 함께 등록한 후 파서가 XML문서를 Parsing하다가 XML태그를 만나면 그 태그에 대응하는 함수를 호출하는 구조를 가진다. 이 XML Tree Builder에서 IDL을 XML DTD로 변환을 하고, 변환된 XML DTD를 가지고 CORBA 객체에게서 받은 결과 값을 DOM API를 통해서 DOM Tree 형태로 만든다. 이 DOM Tree 형태를 가지고 XML Translator에서는 XML 문서로 만든다.

만들어진 DTD와 XML 문서는 XML Document Manager와 SCM(Schema Creation Manager)에게로 전송된다. SCM에서는 받은 DTD와 XML문서를 가지고 XML 스키마를 생성한다. 생성된 스키마는 XML 문서의 Data Type을 확장하기 위한 것이며, 스키마는 XML Document Manager로 전송된다. 이 XML Document Manager에서는 전송받은 DTD, XML문서, 스키마들을 Database에 저장한다. 그리고 이것들을 Database에 저장과 동시에 Web Manager에게 전송을 하고, 최종적으로 Web 클라이언트에게 HTTP 프로토콜을 이용해서 전송이 되어 클라이언트의 Web 브라우저에서 보여지게 되는 것이다. [7,8]

이 구조에서 제일 중요한 것은 CORBA IDL을 DTD로 표현하는 XML Parsing Manager이다. DTD는 객체지향 환경에서 어떤 응용프로그램과 통신을 할 때 Message 형식으로 쓰이기 때문이다.

III. 시뮬레이션

본 절에서는 CORBA와 XML 연동을 위한 응용 웹 서버에서 제안한 Servlet구조와 CGI(Common Gateway Interface)구조와, Applet을 사용한 구조에 대해서 큐 delay를 측정 비교하였다.

시뮬레이션은 OPNET simulator를 사용하였고, Simulation 환경과, 모듈구성은 다음과 같다.

- 언어 : C++, Java
- 코바 제품 : VisiBroker for java 4.5
VisiBroker for C++ 4.51
- O/S : Windows 2000 Server
- CPU : Pentium II 350 Mhz
- RAM : 448 Mhz

- 시뮬레이션 시간 : 7시간
- 클라이언트수 : 1000 user
- 패킷 생성 모듈 : Packet Interarrival Time : exponential(1.0)(s)
 Packet Size : exponential(1024)
 Start Time : 1.0(s)
- 큐 모델
 - ServiceRate
 - Servlet : 9,061 CGI : 5,094 Applet : 2,860
 - Server : 3 개
 - Subqueue : 2 개

비교대상인 Servlet, CGI는 서버 구동구조이며, Applet은 클라이언트 구동구조이다. 이 세 가지 구조의 ServiceRate는 한 개의 프로세스당 처리시간으로 하여 돌출해낸 것이다.



그림 2 Applet을 이용한 구조의 큐 Delay

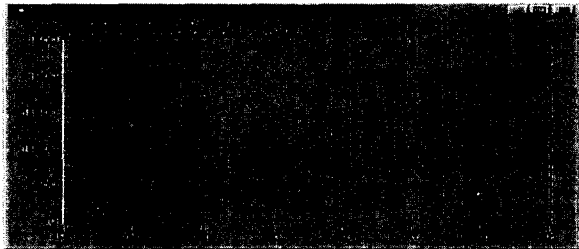


그림 3 CGI를 이용한 구조의 큐 Delay

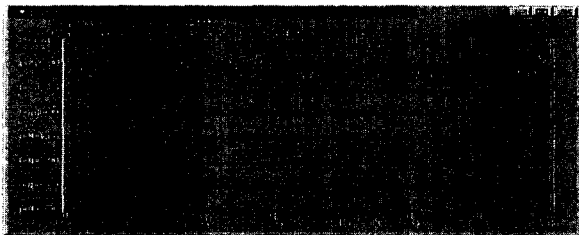


그림 4 Servlet을 이용한 구조의 큐 Delay

본 시뮬레이션에서 ISAPI 구조는 MicroSoft 플랫폼에서만 사용되고, NISAPI는 코바모듈을 지원이 약하기 때문에 제외시켰고, 시뮬레이션 결과는 큐의 Delay와 시간당 평균 Delay를 측정하였다.

그림 2,3,4에서 나타난 것과 같이 각각의 구조에 따라서 차이가 많음을 알수가 있다. 그림 2의 Applet 구조는 클라이언트로

CORBA 모듈을 다운로드 하기 때문에 큐의 Delay가 상당히 큼을 알 수가 있다. 그림 3의 CGI구조도 Applet구조 보다는 낮음을 알수 있으나 그래도 Delay가 크다. 그러나 그림 3의 Servlet구조는 큐의 Delay가 안정적임을 볼 수가 있다.

V. 결론 및 향후 과제

XML은 자기 설명적 데이터이고, 시스템에 독립적이고, 벤더(Vendor) 독립적인 개방된 표준이라는 점에서 웹 환경에서 많은 관심을 받고 있다. 그리고 분산 환경 시스템에서의 CORBA의 다양한 서비스들이 개발되고 있다. 이러한 CORBA 서비스들을 웹 환경도 이용할 수 있는 시스템을 제안하고, 안정성을 입증하였다. 이 시스템은 응용 웹 서버측의 네트워크 오버헤드를 줄일 수 있고, 모든 Java API들을 사용할 수 있기 때문에 CORBA 서버객체의 다양한 서비스들을 지원할 수 있다. 그러나 Database의 DTD, XML 문서와 CORBA IDL의 철저하게 관리를 하여야 될 것이며, CORBA 클라이언트는 Web 클라이언트의 요청이 증가하면 응용 웹 서버에도 어느 정도의 부하량이 증가할 수 있기 때문에 CORBA 클라이언트기술과 XML-RPC[7] 기술인 SOAP(Simple Object Access Protocol)[8,9]을 이용한 기술과의 비교평가가 필요하다.

참 고 문 헌

- [1] 이호섭, 홍충선, "분산환경에서의 CORBA와 XML의 연동구조", 한국정보과학회 춘계학술대회 논문집(A) Vol28 No.1, pp 424-426, 2001. 4.
- [2] 이호섭, 홍충선, "CORBA와 XML 연동을 위한 응용 웹 서버 구조", 한국통신학회 하계종합학술발표회 논문집 Vol23 No.2, pp 1619-1622, 2001. 7.
- [3] XML specification, Level 1, W3C, "http://www.w3.org/TR/1998/REC-xml-19980210"
- [4] CORBA Specification, Version 2.3.1, OMG, "http://www.omg.org"
- [5] Document Object Model, Level 1, W3C, "http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001"
- [6] Document Object Model, Level 2, W3C, "http://www.w3.org/TR/1999/REC-DOM-Level-2-19990923"
- [7] The XML RPC Specifications, UserLand Software "http://www.xml-rpc.com/spec"
- [8] Simple Object Access Protocol(SOAP)1.1, W3 Consortium, "http://www.w3.org/TR/SOAP"
- [9] Simple CORBA Object Access Protocol(SCOAP), OMG, "ftp://ftp.omg.org/pub/docs/orbos/00-09-03.pdf"
- [10] CORBA, XML And XMI Resource Page, OMG, "http://www.omg.org/technology/xml/index.htm"