计算机科学1999Vol.26№.12

CORBA，Java 和 Object Web

CORBA，Java and Object Web

李 琪 梁阿磊 白英彩

（上海交通大学计算机科学与工程系 上海 200030）

Abstract The rapid growing Web has spread to influence many facets of corporate and individual life, but HTTP/CGI are not qualified for the task involved in multi-interactive conversation, trans-area, distributed computing. CORBA can make up the shortcomings of HTTP/CGI, and it cooperates with Java succeesfully. They promise to create real Object Web. The three-tier client/server and interactive sessions used by Object Web and the emerging problems are described in this paper.

Keywords Java，CORBA，Object Web，Client/Server

一、不断面临挑战的 Web

Web 在我们这个信息社会中扮演着愈来愈重要的角色，Internet、Intranet 和 Extranet 是 Web 的三种主要表现形式，已对企业和个人生活的各个方面产生了巨大影响。一个企业在未来市场竞争中获胜的关键是充分利用 Web。Web 的影响面和期望值如此之大，它承担的任务也日趋复杂和多样化，这是 Web 创始者所始料未及的，我们必须对 Web 作出改革。

Web 诞生之初只是一种单向传播静态电子文档的传输媒介，本质上就是一个巨大的基于 URL 的文件服务器。后来，也只不过是用它来支持完成一些简单的、由客户驱动的应用。但是现在情况大有不同，人们需要用 Web 来完成一些大范围的、跨企业的分布式计算，其中涉及更多的客户/服务器交互过程，比如商业应用，更是以多步骤的商业交易活动为代表。总之，Web 已逐渐从一种面向信息的传输介质向面向分布式处理介质转变。

1995年，Web 采用了 CGI（公共网关接口），它遵循一种三层客户/服务器模式。CGI 允许基于服务器的脚本（script）接受由客户通过 HTTP 递交的数据，处理该数据并把结果返回给客户。但是 HTTP 和 CGI 的结合并没有给 Web 性能带来多大起色，反而使 HTTP 变得臃肿、速度慢且不支持状态保存。所以用 HTTP/CGI 创建的 Web 对互操作性支持不够，服务器负载平衡功能不足，无法在跨阶段之间维持一个持续的状态。虽然许多厂商对 HTTP 和 CGI 都作过改进和扩充，但都见效甚微。Web 需要新的通讯模式。

这时面向对象技术似乎为 Web 走出困境点燃了希望之火。Java applet 为创建 Object Web 迈出了第一步。这种功能强大、安全性高并且独立于计算机结构的编程语言，很快就流行起来。但是光凭 Java 还不能完全解决 Web 的性能需求，它需要一个能在网络中任意两个对象之间建立起连接的低层结构——分布式对象平台，这就是 CORBA（公共对象请求代理结构）的任务，Java 和 CORBA 的结合才为创建一个开放异构式 Object Web 提供了可能。

二、CORBA/Java 能为 Web 带来什么？

CORBA 是一种与位置和操作系统无关的分布式对象作用机制，它能够取代今天 Web 使用的 HTTP/CGI，并为 Java 带来三个重要的益处：

1. 出色的性能和灵活性：CORBA 允许客户直接激发服务器上的方法，通过预先编译好的或动态生成的存根（stub）传递参数。这样的网络系统要比 HTTP 支持的系统灵活得多，因为客户构件不只限定于调用 HTTP 支持的预编译好的方法，任何一个 IDL 定义的方法都可以被调用。而且，CORBA 允许传递任何类型的参数，而 CGI 只接受字符串参数；CORBA 不必为每个客户请求都重新运行一下程序，并能存储客户调用状态，而 CGI 对每个客户请求都要重新运行一遍程序并且不支持状态存储；

2. 增容性好：CORBA ORB 可以将收到的客户请求发送给它选择的服务器，其主要目的是平衡服务器负载，CGI 应用却无法将客户请求转发给其它服务器，一个 CGI 应用必须处理它所接收到的所有请求；

3. 底层构件桥梁：Java 应用不能向远程的 Java 对象发出请求。但使用了 CORBA 之后，不仅能实现 Java

应用之间的通讯，还能在用不同语言写成的对象之间实现通讯。

CORBA 还可以将 Java 构件按功能划分成客户端和服务器端构件，这样用户只需下载客户端构件即可，大大减少了下载时间，使得 Web 的客户服务器模式更加吸引人。

CORBA 的设计初衷就是支持三层客户/服务器系统，所以它是 Java 对象模型的自然延伸，为 Java 对象增添了强健的分布式服务。反过来，Java 也对 CORBA 的性能增色不少。

1. 简化了代码分布：根据下载命令，可以很方便地从中央服务器完成代码配置工作，无需系统管理员手工更新每一台工作站；

2. 代码迁移：Java 的移动代码功能使任何一个方法操作可以在系统的机器之间或是在客户服务器之间动态迁移；

3. 代理：我们假定在每一台能被漂浮代理访问的机器上都装有 Java 虚拟机，那么需要代理的 CORBA 应用就能够使用 Java 的可移动代码系统来迁移程序行为，就是说由 CORBA 提供持续的状态存储，Java 实现程序行为；

4. 卓越的语言特色：Java 支持多平台多线程编程，自动垃圾收集和错误处理，这些都非常适合于编写客户/服务器程序。

CORBA2.0定义了 IIOP(Internet Inter-ORB 协议)，能使 ORB(对象请求代理)之间通过 TCP/IP 建立通讯，这样 CORBA 能与 HTTP 同时运行在一个网络中。CORBA 和 Java 相互补充，前者实现网络透明性，后者完成功能实现的透明性。两者结合在一起，就可以让网络上的任何两个对象在任何环境中实现通讯。

三、Object Web 的三层结构

HTTP/CGI 还不能在短时间内完全被代替，而且 HTTP 也有自己的优点，它善于处理非结构化的数据，能轻易地把大量不同类型的数据和信息源组织起来，以简单统一的方式供最终用户访问；CORBA 适合处理结构化的数据以及逻辑复杂的分布式交互操作，把它们结合起来，以 Web 作为纽带，就能实现更好的分布式服务。

这种综合了 HTTP/CGI 和 CORBA/IIOP 的 Object Web 也是三层结构，但与传统的 Web 三层客户/服务器结构又有所不同，这三层结构是：

1. **客户终端**：即第一层，包括传统的 Web 浏览器和新型 Web 桌面系统。与静态的 Web 页面不同的是，新页面中会包含许多真实世界中的对象，具有更逼真的外观和行为特性，这种动态效果是由内嵌在可装卸容器(container)中的 JavaBean 来实现的，可装卸容器可以是 HTML 页面或 Jar，用户可以通过拖拉-放开动作与对象交互，客户的 Bean 程序可以与其他客户或服务器的 Bean 程序交互，而服务器 Bean 程序可以通过 CORBA 事件和回调(callback)来调用客户 Bean 上的方法，另外，HTTP 和 CORBA 同时运行时，HTTP 用来下载 Web 页面、Jar 和图像，CORBA 则负责客户到服务器和服务器到客户的通讯。

2. **中间层**：该层既能为 HTTP 客户也能为 CORBA 客户提供服务，它运行在服务器上。几乎所有服务器的操作系统平台都支持 CORBA 和 HTTP 的结合，比如 Unix、NT、OS/2、NetWare 等。CORBA 对象最终要被打包成企业版的 JavaBean—当作中间层应用服务器，它们将应用要处理的事件逻辑和处理规则封装在对象内部。这些对象就通过 CORBA/HTTP 与客户的 JavaBean 交互。分布程度不高的应用可以通过运行在 HTML 页面中的脚本调用这些对象。

在服务器端的 CORBA 对象同样可以通过 CORBA ORB 实现通讯，也可以通过 JDBC 或其他中间件与现有的第三层中的服务器应用通话，甚至可以使用 CORBA/IIOP 的服务器中枢干线作为通用数据线。

第二层还必须包括一个构件协调器，或者称作对象 TP 监视器，这些构件协调器就是建筑在 ORB 上的 TP 监视器，其作用是管理网络上的对象而不是过程。构件协调器预先启动对象池、分布负载、提供容错处理、以及协调多构件事务。没有这些构件协调器，你就无法管理上百万个的服务器端对象，这是 Object Web 的重要技术。基于 CORBA 的构件协调器产品有 IBM 的 Component Broker，BEA 的 Tuxedo/Iceberg。面所谓的服务器端构件是一个 CORBA 服务器对象，它能实现一最小集合的构件服务。

在 CORBA/Java Object Web 中，构件标题、HTML 页面也存储在第二层中。存储介质可以用可迁移的 Java Jar，其管理由 ODBMS 或 DBMS 完成。

3. **后端**：任何 CORBA 对象可以访问的目标都属于第三层。其中包括过程的 TP 监视器、面向消息的中间件、DBMS、Lotus Notes 和 E-mail 等。

在这样的 Object Web 上，客户与服务器的交互过程可以分为以下五个阶段：

①Web 浏览器下载 HTML 页面。页面中包含了对内嵌 Java applet 的参考。

②Web 浏览器从 HTTP 服务器上获取 Java applet，HTTP 服务器将 applet 以字节码形式下载下来传送给浏览器。

③applet 装载进入 Web 浏览器内存。

④ applet 调用 CORBA 服务器对象。在 Java applet 中能包含由 IDL 生成的客户存根，这使得 applet 能调用 ORB 服务器上的对象。这个阶段会一直持续到通讯双方中的任一方决定断开连接。

⑤ 服务器对象可以有选择地为客户产生一个 HTML 页面。在 Object Web 中一般不需要服务器端的动态 HTML 页面生成，因为一个客户应用被打包成一个 HTML 页面，其中嵌有象 applet 或 JavaBean 这样的构件，与 HTTP/CGI 相反的是，CORBA 可以让你立即与服务器发生交互，只需要点中 HTML 页面上嵌有的任何一个构件，而无需转出页面以获得响应。

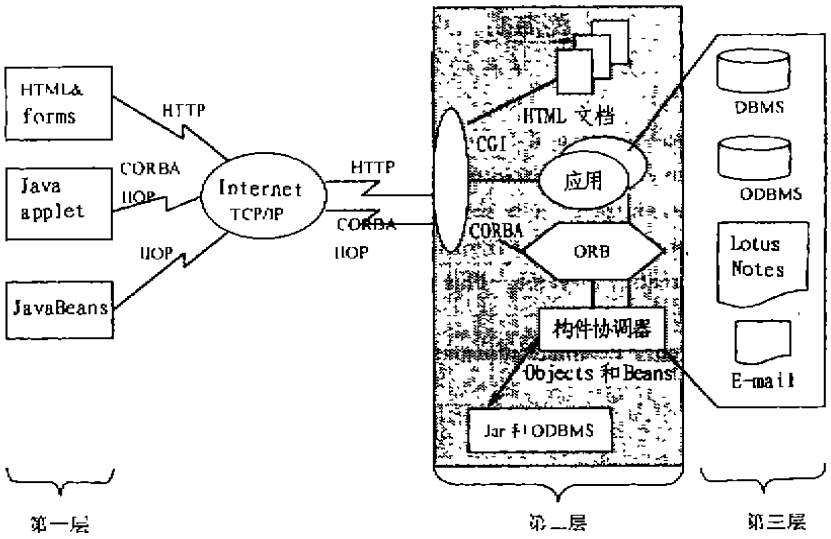

图1 Object Web 的三层结构

四、亟待解决的问题

虽然 CORBA 和 HTTP 可以同时在网络中使用，但在今天这个基本使用 HTTP 的 Web 上加入一个新的通讯协议 IIOP 并非畅通无阻，目前有以下几个问题亟待解决：

1. 防火墙的阻挠：因为防火墙不能识别 IIOP 协议，必须先求助于 HTTP 管道，将 IIOP 包转换为 HTTP 包，才能继续通讯。这样使用 IIOP 协议的应用程序就受到性能限制；

2. 对 IDL 语言的全面支持：目前只有少数语言支持 IDL 的自动编辑，在其他语言中只能手工完成此项任务；

3. 网络带宽的限制：那些会造成瓶颈效应的网络应用仍然会受到网络带宽的限制。

参 考 文 献

1 Object Management Group, Common Object Request Broker Architecture Specification, Version 2.0, 1994

2 Horton, Beginning Java, Wrox Press Ltd., Birmingham, U.K., 1997

3 Stanek W R. Distributed Programming with CORBA and IIOP. PC Magazine, 1997(Sept.): 241～243

4 Rubin K. The Future of Objects. J. Object-Oriented Programming, 1997, 10(4): 15～18

5 Dreyfus P. Corba: Theory and Practice, 1997. Available at: http://developer.netscape.com/news/viewsource/dreyfus_corba.html

6 Garg R. Corba: The Future of Web Computing, 1998. Available at: http://www.sun.com/developers/devnews/summer97/corba.html