计算机科学2000Vol 27№. 6

一种全新的 Java 计算模型*)

A New Java Calculating Model

李 钢 李增智 张 鹏

(西安交通大学系统结构与网络研究所 西安 710049)

Abstract Starting from current prevalent calculating model in developing Java application, this paper analyzes some defects existing in this model and gives corresponding solutions. At last, we give a multi-tier service-based calculating model, which can be well used for reference in developing big-scale Java application.

Keywords Java, CORBA, Servlet, XML, DOM

1 引言

随着 Internet 和 WWW 的流行，Java 已经成为开发应用系统的首选编程语言之一。当前，Java 应用程序的开发大都采用所谓的三层结构：即客户层、事务逻辑层和数据库层，如图1所示。其中，客户层负责为前端用户提供图形用户界面；事务逻辑层则处理所有的服务访问和数据库访问；数据库层则提供数据存储的功能，即保存数据本身。

图1 Java 应用程序的三层计算模型

为了使客户端尽可能"瘦"，人们在客户层选用 Java Applet 和 HTML，这样客户端程序就可以被动态下载到任何支持 Java 的机器上。在事务逻辑层，人们使用 Java 开发可以独立运行的应用程序。由于这两层都采用了 Java 作为程序开发语言，所以可使用 RMI 作为这两层之间的通信机制。最后，中间层与数据库层之间的通信是通过 JDBC API 以及第三方厂家的 JDBC 驱动器共同来完成的。该模型实现起来简单，而且结构清晰，易于维护，但是还不乏改善之处。

2 计算模型的改进

下面我们从以下几个方面对该模型加以改进。

a. 分布式通信机制的转变 在当前结构中，我们使用 RMI 作为客户端与服务器之间的通信机制。虽然 RMI 是标准 JDK 的一部分，使用起来简单，但我们将采用 OMG(对象管理组)的 CORBA(公用对象请求代理体系结构)来代替 RMI，这其中的重要原因就是考虑到系统的可扩展性。大家知道，RMI 是用于分布式环境的一种通信机制，但它所连接的两端系统必须都是由 Java 实现的，而 CORBA 则不然，它是支持异构系统的分布式框架，换句话说，CORBA 的主要目的是为软件开发人员提供不依赖于单个操作系统/编程语言的系统开发途径，这些系统可能硬件不同，也可能是操作系统和编程语言不同，这样，一个用 C 语言实现的服务器就可能嵌入到我们的系统中，从现今来看，已有多家公司发行了 CORBA 的产品，在最新的 JDK2.0 中就含有100%纯 Java ORB、命名服务并能获得相应的 Java IDL 编译器。

b. 使用 Java Web 服务器和 servlet API 当前结构中的事务逻辑层是通过编写可以独立运行的 Java 程序来完成的，该程序将负责用户认证，并提供安全保障等功能。使用 JavaSoft 提供的 Java Web 服务器和 servlet API 后，该层的开发任务将被大大地简化。

所谓 servlet，就是服务器方的 Java 软件，它能够运行在任何实现 servlet API 的 Web 服务器上，而 Java Web 服务器则是由 JavaSoft 开发的支持 servlet API 的 Web 服务器(当然，此处可以用任何实现 servlet API 的 Web 服务器来代替 Java Web 服务器)。客户端是通过 URL 来使用 servlet 的。从动态创建

*)受国家863重点项目(863-511-946-008)的资助。李 钢 博士生，研究方向：Web 技术在 TMN 中的应用，网络协议工程。李增智 教授，博导，研究方向：计算机网络及应用，电信管理网和电子数据交换 EDI。

HTML 页面到处理各种数据库业务.servlet 几乎无所不能。更为重要的是,使用 servelt 可以很容易地实现管理多线程、处理客户请求以及提供安全保障等功能,从而使开发人员将主要精力放在完成特定的服务功能上。

c. 临时服务器层的引入　在当前的三层模型中存在两个不足。首先是 Java applet 的下载时间有时很长。由于 applet 是在运行时由应用服务器动态下载到客户端,当两端所在机器距离很远时,applet 可能需要几分钟的下载时间。我们需要某种机制能将 applet 和静态的数据缓存到离客户很近的地方。其次是关于访问网络资源,如文件和打印机等的问题。由于 Java 沙箱安全模型的限制,applet 不能访问任何本地和网络资源,也不能和除了从其下载的那台机器之外的任何其它机器通信,所以文件访问和打印等工作都必须在事务逻辑层完成。这意味着信息可能从客户端被发送到事务逻辑层,然后又被反馈给在客户端附近的文件服务器。显而易见,这些都是非常费事而又低效的方法。我们需要某种机制能够对网络资源进行有效的访问而又绕过 Java 安全模型的限制。

为此,我们引入新的一层,称为临时服务器层。其中,临时服务器是一台运行 Java Web 服务器的机器,它位于它所服务的客户的附近。该层的主要功能包括:作为 applet 和静态应用数据的高速缓存;驻留一些用 servlet 实现的可以访问网络资源的服务。一般而言,这些网络资源都位于用户附近;此外,临时服务器层还将帮助客户寻找他们需要的服务。

d. 定义可重用服务模块　最终,我们扩展了代码重用的概念。此时,应用系统将由提交服务请求的客户程序和许多完成各种服务功能的可重用模块构成。这些服务包括用户认证、打印、文件访问以及数据访问等。这些服务模块将被设计为可在多个应用之间共享使用。

e. 构建可适应性的服务器　对于服务器方程序而言,数据信息的动态扩展性是至关重要的,这源于以下原因:

·数据和操作应该分开:不应该将系统处理的各种数据"硬"编码到程序内部。相反,系统应该能够以一种可处理多种数据类型的通用方式来处理数据。

·可伸缩性:即服务器中的不同组件在处理信息上应具备高度的灵活性。

·可适应性:服务器应该能够很容易地被集成到各种不同的环境中。

·可维护性:通过"高度的适应性",服务器的维护和错误检查应变得更容易。

为了做到这一点,我们引入 XML 和 DOM 技术。XML,即扩展性标记语言,是由 World Wide Web 联合组织(W3C)认可的标准,负责定义文档的结构,实际上,我们可将一个 XML 文件看成是一个树状的数据库,并可以在其上进行各种数据查询操作;W3C 的另一个技术就是 DOM,即文档对象模型(DOM),它将允许 HTML 和 XML 文件在程序控制下访问结构化数据。这两种技术的引入,克服了传统面向对象编程中的缺点,即数据和操作不能分离。

3　基于服务的多层计算模型

针对当前模型的种种改进最终导致了一种全新的计算模型,即基于服务的多层计算模型,如图2所示。其中,客户层被编写为一个 applet,可运行在任何支持 Java 运行环境的 Web 浏览器中,所以该层是"瘦"客户。客户层不直接和数据库打交道,而且也不保留任何局部状态信息。

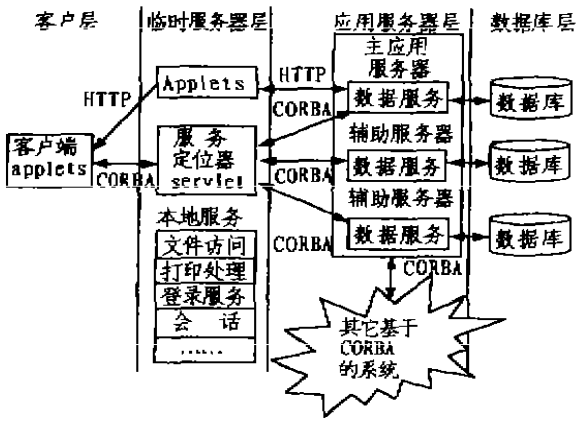

图2　基于服务的 Java 应用程序多层计算模型

临时服务器层包括一个用 servlet 实现的服务定位器,客户层通过它可以找到相应的提供该服务的模块。例如,若客户端程序需要打印,它必须首先通过临时服务器层的服务定位器获得打印服务模块的"句柄",使用该"句柄"就可以得到最终的打印服务。实际上,临时服务器层就象一段桥梁,将客户层 applet 和所需的可重用服务模块联系起来。

应用服务器层提供对数据的访问并实现事务的逻辑处理功能。

在临时服务器层和应用服务器层的各个服务模块中,我们都使用了 XML 和 DOM 技术,以使得功能处理和应用数据完全分离。

数据库服务器层负责保存应用数据,有时,它将与临时服务器和应用服务器层共同完成事务处理、数据

有效性检验以及永久存储等功能。

客户层和临时服务器层之间的通信是通过 HTTP 和 CORBA 来完成的，即利用 HTTP 调用相关的 servlet，从而得到所需的 CORBA 对象的引用，然后就可以使用 CORBA 机制了。相同的机制也适用于临时服务器层与应用服务器层之间的通信。而应用服务器层和数据库层之间的通信则是通过 JDBC 来完成的。

由于采用 CORBA 作为通信机制，在未来，我们也可以将其它使用 CORBA 的服务器（如用 C++ 实现的服务器）结合进我们的系统。

4 运行过程

当用户第一次访问应用程序时，相关的 applets 和 servlets 将从应用服务器层被下载到临时服务器层，对最终用户而言，applet 是从临时服务器直接下载的。一旦 applet 最终下载到客户端，客户就可以向临时服务器上的某个 servlet 发出服务请求（必要时，servlet 可以从应用服务器动态下载）。

servlet 接收到客户的请求后，或者在本地进行处理，或者将该请求再发送到应用服务器上。一般来说，诸如打印和文件访问之类的请求由临时服务器上的服务模块处理；而应用数据的处理请求则发送到应用服务器上。该模型有如下优点：

a）高度的可扩展性：采用 XML 和 DOM 技术使程序中的处理功能和应用数据完全分离；通过中间各层使前端客户不直接和数据库打交道以及基于 CORBA 的分布式处理环境都大大增加了系统的可扩展性。

b）大多数逻辑事务处理都在应用服务器和数据库层完成，所以客户是"瘦"客户。

c）充分利用了网络资源。对于一些静态的应用数据，可以直接缓存到临时服务器上。

d）显著地提高了系统的效率。这表现在以下几个方面：applet 下载时间减少、打印和文件访问之类的服务就可直接在离客户最近的临时服务器上实现、服务模块都是可重用模块、不影响客户就可以修改应用服务器。

e）因为客户端 applet 和服务器方的 servlet 都可以被动态下载，所以系统的安装将大大简化。

结束语 由于采用了最新的一些相关技术并借鉴了高速缓存的思想，使得基于服务的 Java 多层计算模型在系统开发效率、系统运行效率以及系统的未来扩展性方面都比原先的三层计算模型有了较大的提高。随着 Java 技术、网络技术和其它相关技术的进一步发展，Java 计算模型还会有进一步发展。至少从现在来说，我们可以预见引入 JavaBean 技术将是该计算模型进一步发展的必然趋势。

参 考 文 献

1 Available at: http://java.sun.com/products/jdk/2/docs/guide/extensions/spec.html
2 Available at: http://java.sun.com/products/jdk1.1/docs/guide/rmi/spec/
3 Available at: http://java.sun.com/products/java-server/servlets

（上接第39页）

定理7 U(X)=U(Y)的充要条件是它们有相同的上粗元。

证明：由定理6即可得证。

定理8 设$[X]_\approx$和$[Y]_\approx$是两个粗集，则：(1)$Z\in_U [X]_\approx \cup_\approx [Y]_\approx$当且仅当 $Z\in_U [X]_\approx$或 $Z\in_U [Y]_\approx$；(2) $Z\in_U [X]_\approx \cap_\approx [Y]_\approx$当且仅当 $Z\in_U [X]_\approx$且 $Z\in_U [Y]_\approx$

证明：由定理6和粗交、粗并的定义即可得证。

5 上粗元、下粗元与粗元的关系

定理9 Y 是$[X]_\approx$的粗元当且仅当 Y 是$[X]_\approx$的下粗元和上粗元。

证明：由定理2、定理3和定理6即可得证。

定理10 $[X]_\approx=[Y]_\approx$的充要条件是它们有相同的粗元。

证明：由定理4和定理7即可得证。

定理11[7] 设$[X]_\approx$和$[Y]_\approx$是两个粗集，则：(1)$Z\in_\approx [X]_\approx \cup_\approx [Y]_\approx$当且仅当 $Z\in_\approx [X]_\approx$或 $Z\in_\approx [Y]_\approx$；(2)$Z\in_\approx [X]_\approx \cap_\approx [Y]_\approx$当且仅当 $Z\in_\approx [X]_\approx$且 $Z\in_\approx [Y]_\approx$。

证明：由定理5、8、9和10即得证。

结论 本文对文[1]进行了更深入的讨论，拓广了粗集中的粗元概念，通过引入的上粗元和下粗元概念，对粗集的基本结构元素进行了细致的描述和刻画。

参 考 文 献

1 Bonikowski Z. Algebraic Structures of Rough Sets, Rough Sets, Fuzzy Sets and Knowledge Discovery. Springer-Verlag, 1994
2 Bonikowski Z. A Certain Conception of Calculus of Rough Sets. Notre Dame Journal of Formal Logic, 1992, 33
3 Bryniarski E. A Calculus of Rough Sets of the first order. Bull. Pol. Ac.: Math., 1989, 37
4 Gehrke M., Walker E. On the Structures of Rough Sets. Bull. Pol. Ac.: Math., 1992, 40
5 苗夺谦，王珏. 粗糙集理论中知识粗糙性与信息熵关系的讨论. 模式识别与人工智能，1998, 1
6 王志海，等. 基于粗糙集合理论的知识发现综述. 模式识别与人工智能，1998, 2
7 何华灿，等. 经验性思维中的泛逻辑. 北京：中国科学(E辑)，1996, 1
8 何华灿，等. 人工智能导论. 西安：西北工业大学出版社，1988