

# WEB应用服务器体系结构研究及其应用

胡泳 张志浩 陈福民 (上海同济大学计算中心 200092)

**摘要:**本文对WEB应用服务器的功能和体系结构进行了比较深入的探讨和研究,并对具有代表意义的ORACLE WEB APPLICATION SERVER (OWAS)进行分析、探讨。

**关键词:**C/S结构 B/S/S结构 WEB应用服务器 OWAS 分布式组件模型

## 一、引言

传统的企业计算模型——两层的C/S结构已经不能适应日益增长的网络计算的需要,INTERNET的出现为IT管理者和开发者提供了一个崭新的思路:将应用程序和计算逻辑全部移植到服务器端,客户端只保留界面统一的浏览器,形成了B/S/S型的三层结构。统一的界面,方便的管理,JAVA语言的出现更使其如虎添翼,B/S/S结构几乎成了众望所归的理想型计算模式。然而随着网络规模的日益扩大,网络功能要求的逐步加强,这种B/S/S结构也显露出它的不足。

1. WEB服务器的负担越来越重,请求响应速度逐渐减缓,速度上与传统C/S结构相差明显。

2. 传统C/S结构应用程序与B/S/S结构应用的结合不紧,但为了保护以前的投资,只好使用C/S、B/S/S两套系统,给程序开发、维护带来诸多不便。

3. 虽然WEB访问数据库服务器或文件服务器的方式有多种,如CGI,API,JDBC等,但是它们都没有同时解决开发效率和系统性能两方面问题。

4. WEB服务器对将要接到的请求数不能有比较准确的预期估计,因此实现管理网络状态、平衡负载等对应用程序管理和调节的功能困难很大。

5. 传统C/S结构中的安全机制已经不适应INTERNET/INTRANET模式的要求,必须用新的、基于公共密钥算法的机制取代,例如SLL,这样就增添了开发任务和困难。

6. 与传统分布式组件模型的结合不完美,管理上增加了复杂度。

基于以上原因,在网络为计算的计算环境中,单靠Web服务器并不能使有强大处理能力的分布式应用为大量用户所用,人们通常需要用协议和API作为接口来为应用程序提供所需的扩展,或者创建新的应用程序。

因此迫切的需要一种能适应当前复杂网络计算的平台,WEB应用服务器正是在这种情况下产生的。

## 二、应用服务器功能及体系结构分析

WEB应用服务器是一种支持多种工业标准的协议和界面引擎,它支持的工业标准包括因特网通信协议、数据库互连JDBC和ODBC、分布式组件模型CORBA、EJB等。大部分WEB应用服务器都能提供管理网络状态、平衡负载等对应用程序管理和调节的功能,以及容错性、先进的安全性和对多个数据库连接等功能。一些应用程序服务器还将集成的开发环境加以综合,这意味着企业开发人员可以在单一的平台编译、连接并且配置Web应用程序。

WEB应用服务器是在传统的Web服务器与数据库或者传统应用程序之间的一个中间层。虽然传统WEB服务器可以通过小程序和插件显示静态数据并处理基本的数据库访问,但一个综合的软件引擎要能集成不同来源的数据以作出不同处理,而这正是WEB应用服务器的长处。WEB应用服务器主要与以下三方面的技术相关:

1. 开发环境。企业开发人员需要一种创建新组件,并且将已存在的组件加以集成的开发环境。

2. 应用程序的集成。由于企业的计算环境相当复杂,它综合了传统的应用程序和新式的多层次应用程序,因此开发人员需要集成应用程序,以便创建出比简单地将各部分加以整合更为强大的崭新的应用程序。

3. 应用程序的配置。由于Web应用程序是分布式的,同时其组件运行于不同的服务器上,并且有大量的用户群体对其进行访问,因此这些应用程序就需要一个配置平台的支持,以便在使用剧增时能有效地扩展,并保持系统稳定的性能。

由于 WEB 应用服务器是一个新兴的概念,对它还没有一个完美的定义,但就其实现的功能而言,一般应包含以下几个方面:

- 1. Web 服务器。接收对 HTTP 请求,以静态 HTML 页面或调用中间件生成动态 HTML 页对其进行响应。
- 2. 数据库系统访问。使用中间件实现对分布式数据库进行访问。
- 3. 分布式组件模型及对象请求代理。实现用不同语言编写,存放在不同物理位置上的组件间的通信和协同工作。
- 4. 事务监控。使用先进的资源管理技术,使众多的客户相互间看来与数据服务器都有一对一的连接;同时也进行事务管理,提供扩展性与信赖性。
- 5. 安全和目录控制。调节用户请求和请求的资源,以增强访问策略。

图 1 是一个一般意义上的 WEB 应用服务器的体系结构示意图:

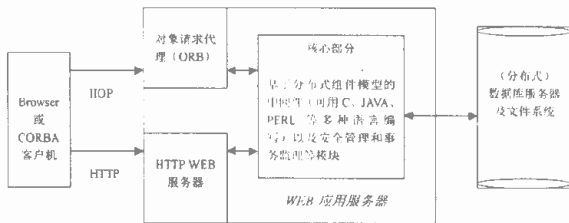


图 1 WEB 应用服务器的体系结构示意图

总体看来,因特网的火爆和网络服务器的普及滋养了 WEB 应用服务器,它是一种将网络、客户机/服务器和主机应用程序联系在一起,并且使应用程序得到最佳运行效率的计算平台,它为要转向 Web 并要保护以前投资的用户提供了一个高性能多线程的环境,是网络计算平台的发展趋势。

### 三、OWAS 体系结构分析及应用

WEB 应用程序服务器的强大优势使业界许多大公司纷纷推出自己的产品,市场目前的形势正是群雄逐鹿、诸侯纷争,不同的公司、不同的产品各有所长。

ORACLE 作为世界上最大的数据库厂商早在 1997 年就推出了其 OWAS3.0 版,98 年末,又推出 OWAS4.0 版,它集成了一般意义上 WEB 应用程序服务器的所有

技术,正在成为事实上的最佳配置平台。同时,ORACLE 无论在数据库平台,前台开发工具,还是最终的客户应用方面都有一套完整的解决方案。因此在 OWAS 上建立基于 WEB 应用服务器的企业级网络计算模型可以得到可靠的技术保障。下表列出 OWAS 对一般意义的 WEB 应用服务器各方面技术提供的支持:

WEB 应用服务器体系结构研究及其应用

WEB 应用服务器技术	OWAS 相应的支持
WEB 服务器	自带一个 HTTP WEB 服务器,同时也支持其它的 WEB 服务器,如 IIS、NES 等。支持可管理、编程的 WEB 事务和会话支持 SSL 安全机制
分布式组件模型	包含基于 CORBA2.0 的 ORB 支持标准的 IIOP 协议
安全和目录控制	基本的安全认证(用户名/密码)基于区域和 IP 的认证与 ORACLE 数据库安全认证紧密结合支持将 URL 映射到动态页面的虚拟路径通过控制列表保护虚拟路径的安全访问
数据库访问	通过 OCI(Oracle Called Interface)访问 ORACLE 数据库通过 ODBC、JDBC 访问非 ORACLE 数据库
事务监控	多线程支持增强的请求数等支持两相提交事务支持大的用户池

图 2 是 OWAS 体系结构逻辑示意图:

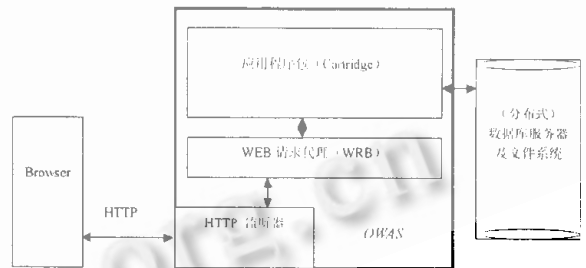


图 2 OWAS 体系结构逻辑示意图

OWAS 逻辑上可以分为三部分:

1. HTTP 监听器。即传统意义上的 WEB 服务器,它负责接受浏览器对静态 HTML 页面和 CGI 程序的请求,返回目标页,其他的请求都交给 WEB 请求代理处理。它可以是 OWAS 本身绑定的 ORACLE WEB SERVER,也可以是其他厂家的,如微软的 IIS,网景的 NES。

2. WEB 请求代理(WRB)。它实现对 Cartridges(应用程序包)请求的管理,包括负载平衡,安全机制,自动纠错,事务服务等。它执行的功能应当与 CORBA 中的对象请求代理,可以与运行在网络上任意主机上的应用程序(在 OWAS 中就是 Cartridges)通信。WRB 可以认为起

到管理整个 OWAS 系统运行的功能。

3. 应用程序包(Cartridges)。相当于 CORBA 中的组件,是服务端的应用,处理来自客户端的 Cartridges 请求,应该说 OWAS 的大部分功能都是由不同的 Cartridge 实现的。它是通过 WRB 与客户端实现通信的。

Cartridges 利用 WRB PAI 创建,通过它可以实现与 WRB 的通信,同时获得一份 WRB 执行机制的拷贝。OWAS 本身自带了一些 Cartridges,包括 JAVA Cartridges, PL/SQL Cartridges, LiveHTML Cartridges, ODBC Cartridges, Perl Cartridges, FORM50Cart Cartridges 等,同时用户也可以用 WRB API 语言编写自己的 Cartridges 而实现特殊的要求。Cartridges 的体系结构如图 3:

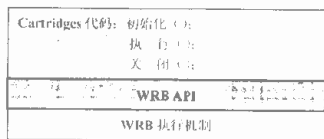


图 3 Cartridges 的体系结构

由于 WRB 的管理,各个 cartridge 是相互独立的,一个 cartridge 运行出错并不影响其他的 cartridge 运行;另外,对于每一个 cartridge 同时可以存在多个实例(Instance),一个实例就是 cartridge 的一个执行过程。下面对 OWAS 处理客户请求的过程作简单的描述:

当 OWAS 接收到客户机的请求时,WEB 监听器首先分析该请求并决定如何处理;如果请求是一个静态文件,监听器就从操作系统中取出该文件并发送给客户机;如果请求是执行一个 CGI 文件,监听器便执行 CGI 脚本并把结果返回给客户;如果请求是一个 cartridge,监听器就把请求转移给 WRB,WRB 查找虚拟路径以确定请求

cartridge 的类型,然后调用认证机制对客户进行认证,最后把请求发送给相应 cartridge 的一个实例,如果该 cartridge 当前无实例,则通知 WRB 创立一个。剩余的工作就是 cartridge 的实例处理请求并把结果返回给客户。例如,监听器收到一个请求:<http://mymachine.mydomain/java/HelloWorld>,由于不是静态 HTML 页或 CGI 文件,请求被传递给 WRB,WRB 分析请求并查找虚拟路径,由/java/知道这是一个 JAVA cartridge 请求,便把请求发送给 JAVA cartridge 的一个实例,该实例分析请求,寻找 JAVA 类文件 HelloWord.class 并装载运行,最后把结果通过 WRB 返回给客户。

#### 四、结束语

WEB 应用服务器是一个很有前途的技术,它适合当今日益增长的网络计算模型需求。然而它毕竟是一个新技术,虽然基于 WEB 应用服务器的网络计算模型为企业的 IT 带来了无限的生机与活力,但是如何构建这样一个模型对广大的 IT 管理者和开发者来说并不是一件容易的事。由于 WEB 应用服务器市场方兴未艾,对它定义还没有达到一致,各厂商都有自己的说法,其产品功能也不尽相同。因此选择什么样的平台,使用那些开发工具,利用什么技术把分布的组件统一为一个整体,以及怎样管理一个庞大的计算模型而使其更安全更有效,所有的这些都需要广大的研究人员去探讨,去研究,去实践。

#### 参考文献

- [1] <http://www.olab.com>
- [2] <http://www.oracle.com>
- [3] 中国计算机报 1993.3.11

(来稿时间:1999年4月)