

# Tomcat 和 IIS 的 Web 服务集成研究

乐挺

(同济大学, 上海 201804)

**摘要:** 目前在企业应用级开发领域, J2EE 应用极其广泛, 在开发中一般采用 Tomcat 作为 Web 服务器, 该服务器能够良好支持 Servlet/JSP, 但和 windows 平台下的 IIS 服务环境兼容性较差。而 windows 平台原有的 IIS 服务器虽然能够良好处理静态页面, 但无法接收 Servlet/JSP 请求。因此这篇文章对 IIS 和 Tomcat 的机理进行了研究, 通过 IIS 本身具有的 ISAPI 筛选器技术, 实现来自客户端请求的重定向, 将 Tomcat 和 IIS 的 Web 服务共同集成在本机上, 以兼顾双方的优势。

**关键词:** Tomcat; IIS; ISAPI; Servlet/JSP; JDK; 集成; Web 服务器

**中图分类号:** TP311 **文献标识码:** A **文章编号:** 1009-3044(2008)36-2761-03

## Research on Web Service Integration of Tomcat and IIS

YUE Ting

(Tongji University, Shanghai 201804, China)

**Abstract:** Nowadays, J2EE is quite widely used in enterprise application development. In this method, developers always use web server "Tomcat", which is capable of Servlet/JSP but not well compatible in IIS environment of Windows platform. On the other hand, IIS service is accomplished in static HTML page but not able to deal with Servlet/JSP request. Therefore, this paper analyzes the mechanism of Tomcat and IIS, discuss about how to make use of ISAPI in IIS to do redirection of Servlet/JSP request which leads to the nice integration of Tomcat and IIS.

**Key words:** tomcat; IIS; ISAPI; servlet/jsp; JDK; integration; web server

## 1 引言

随着时代的进步, SUN 公司的 Java2 Enterprise Edition (J2EE) 网络技术已经成为了目前使用最广泛的 Web 应用开发技术, 它是一种利用 Java2 平台来简化企业解决方案的开发、部署和管理复杂问题的体系结构, 包括客户层技术、网络层技术 (Servlet/JSP)、Struts 架构、服务器层技术 (EJB)、数据事务与安全、打包和部署等, 涵盖的领域十分广阔。而在 J2EE 的开发过程中, 需要一个稳定的应用服务器。在通常的情况下, 我们使用开源服务器 Tomcat 来作为 web (Servlet/JSP) 的开发容器, 这也是 SUN 公司官方推荐的服务器。

而针对 Windows 的系统平台, 微软公司也开发了一款架设 web 服务器的工具 IIS, 即 Internet Information Services (网络信息服务), 这款工具通过超文本传输协议 (HTTP) 传输信息, 并可通过配置来提供文件传输协议 (FTP) 和其他服务, 如 NNTP 服务、SMTP 服务等。

由于是官方开发, 因此 IIS 和 Windows 系统具有天然良好的兼容性, 并可以支持系统中大量的基本网络服务, 如论坛服务。但是因为设计功能的不同, IIS 不支持 Servlet/JSP 机制, 因此不能作为 J2EE 的网络服务容器。在这种情况下, 为了使得 J2EE 开发平台和 windows 的网络服务能够同时共存, 我们可以试着进行双方的配置整合, 将功能有机地集成在一起。

## 2 Servlet/JSP 机制和 Tomcat 服务器概述

Servlet/JSP 是 J2EE 提供的一套跨平台的高效机制, 用来作为客户机/服务器的解决方案, 有着强大而独特的功能。

### 2.1 Servlet 介绍

Servlet 是供服务器端加载的程序, 在客户端通过超文本传输协议 (HTTP) 发出请求后, 服务器端检查对应 Servlet 的状态, 如果尚未被加载, 则服务器将其加载到 Java 虚拟机并且将请求发给该程序, Servlet 接受请求后便执行某种过程作业, 然后向服务器返回相应的结果, 最后, 服务器将结果的应答发送给客户端。

由于 Servlet 程序是在服务器上加载和执行, 客户端浏览器并不直接和 Servlet 通信, 因此就不会出现那些和 Applet 机制相关的安全性问题。而且 Servlet 是用 Java 编写的, 所以一开始就具有平台无关性, 具有一次编写就可以随意移植运行 (Write Once, Run Anywhere) 的特性。除此之外, Servlet 还具有以下几个优点:

运行效率高。在传统的 CGI 中, 每个请求都要启动一个新的进程, 如果 CGI 程序本身的执行时间较短, 启动进程所需要的开销很可能反而超过实际执行时间。而在 Servlet 中, 每个请求只是加载一个轻量级的 Java 线程进行处理, 并且处理请求的数个线程只需要加载一份 Servlet 类代码, 大幅度提高了运行效率。

扩展性和安全性良好。Servlet 程序是用 Java 语言编写的, 因而具备 Java 的所有优点。Java 是一种健壮的、面向对象的编程语言, 很容易扩展以适应各种需求。并且在运行时, 编写好的 Servlet 程序是存在于服务器端的, 因此从外界运行 Servlet 程序的唯一方法是通过服务器调用, 因此在服务器防火墙的保护下, Servlet 程序能够得到高水平的安全性保障。

成本经济, 节省投资。由于 Servlet 技术的广泛使用, 市场上有许多廉价甚至免费的 Web 服务器可供个人或小规模网站使用, 对

收稿日期: 2008-08-02

作者简介: 乐挺, 男, 硕士研究生, 主要研究方向: 软件工程。

于其余的现有服务器,即使它不支持 Servlet 技术,也往往可以免费或通过极少的投资来追加这部分功能。

## 2.2 JSP 介绍

在 Servlet 机制的基础上,J2EE 架构同时提供了一种更加简易的 JSP(Java Server Pages)机制。JSP 是一种脚本语言,它包装了 Servlet 系统的界面,简化了 Java 和 Servlet 的使用难度,同时通过扩展 JSP 标签(TAG)提供了网页动态执行的能力。在运行时,JSP 引擎会按照 JSP 的语法,将 JSP 文件转换成 Servlet 代码源文件,然后调用 Java 编译器,按照正常 Servlet 的方式进行编译,最后在服务器载入运行。

使用 JSP 技术,开发人员无需了解 Servlet 机制,也可以使用 HTML 语法和标签来设计和开发动态网页,并且由于 JSP 引擎的自动机制,JSP 页面在修改之后能立刻看到更新结果,而无需手动重新编译,这样大大地简化了网页的开发流程。同时 JSP 页面也具有 Java 的语言特点,几乎可以运行于所有平台,具有极其广泛的市场。

## 2.3 Tomcat 服务器介绍

Tomcat 是 SUN 公司官方推荐的 Servlet/JSP 服务器。它是一个免费开源的 Servlet 容器,也是 Apache 组织的 Jakarta 项目中一个核心项目,不但具有轻量级和灵活嵌入系统的优点,并且可以支持 Servlet/JSP 的最新规范,因此得到广泛的应用。

在安装 Tomcat 服务器之前,系统必须先安装 JDK,即 Java Develop Kit(Java 开发工具包),这是 SUN 公司免费提供的 Java 语言的开发包,包含了编写、运行 Java 程序所需要的全部工具。

## 3 Tomcat 和 IIS 的集成配置

Tomcat 本身具有 Web 服务器的功能,因此可以单独运行,但 Tomcat 处理静态 HTML 页面的速度较慢,并且和目前 Windows 系统的 IIS 环境下开发的程序兼容性较差,而 IIS 服务器则不能解析 Servlet/JSP 页面,因此我们将 Tomcat 和 IIS 集成在同一个 Web 服务器上,这样既能支持强大的 Servlet/JSP,又能完全兼容原有 IIS 环境下开发的各种软件。

### 3.1 安装 IIS

如果使用的 Windows 系统是服务器(Server)版本,如 Windows 2000 Server,大多已经自动安装了 IIS,因此可以略过这一步。如果使用的是其他版本的 Windows 系统,则需要首先安装 IIS。安装的过程非常简单,将 Windows 系统的安装盘插入光驱中,然后在系统桌面的控制面板中运行“添加删除程序”,选择“添加/删除 Windows 组件”,在打开的对话框中选中“Internet 信息服务(IIS)”,然后单击“下一步”即可进行安装。注意不同 Windows 系统的安装盘的 IIS 版本可能不同,比如 Windows 2000 安装盘中是 IIS 5.0,Windows 2003 安装盘中是 IIS 6.0。

### 3.2 安装和配置 JDK

JDK 可以在 SUN 公司的官方网站上免费下载,目前最新的 JDK 1.6 版本的安装程序名称为 jdk-6-windows-i586.exe,点击运行即可进行安装。接下来需要设置环境变量,在桌面的“我的电脑”图标点击鼠标右键,依次选择“属性”——“高级”——“环境变量”,在上方用户变量的窗口中点击“新建”,新建一个名称为 JAVA\_HOME 的变量,值为 JDK 的安装目录(默认为 C:\Program Files\Java\jdk1.6.0)。然后再按同样方法新建一个名称为 CLASSPATH 的变量,值为 JDK 安装目录下 lib 目录下 dt.jar 文件的地址(默认为 C:\Program Files\Java\jdk1.6.0\lib\dt.jar)和 tools.jar 文件的地址(默认为 C:\Program Files\Java\jdk1.6.0\lib\tools.jar),两个地址之间以分号隔开。最后在用户变量窗口中选择系统变量“PATH”,点击“编辑”,在该变量值的尾部添加 JDK 安装目录下 bin 目录地址(默认为 C:\Program Files\Java\jdk1.6.0\bin)。完成这些环境变量的设置后,可以试着在命令行(运行——cmd)窗口输入“java”,如果出现 Java 命令参数等信息,则表明 JDK 的安装和配置成功。

### 3.3 安装 Tomcat

Tomcat 服务器的安装程序可以在 apache 组织的官方网站上免费下载,目前最新的 Tomcat 6.0 版本的安装程序名称为 apache-tomcat-6\_0\_10.exe,点击运行即可进行安装。安装完毕后启动 Tomcat 服务进程,然后在浏览器(如 Internet Explorer)中输入“http://localhost:8080/”进行测试,如果出现 Tomcat 的默认页面,则表示 Tomcat 服务器和 JDK 的安装配置已经成功。

### 3.4 获取 JAKARTA 连接器

为了能够集成 IIS 和 Tomcat 服务器,我们可以使用 IIS 服务管理器中的 ISAPI 筛选器,该筛选器是 IIS 对外的一个接口,可以调用各种扩展插件。我们可以在接口上安装一个 Tomcat 重定向插件,这样 IIS 在每次得到 Servlet/JSP 请求时,ISAPI 筛选器就能通过 ajp13 协议,调用外部的该插件,对 Servlet/JSP 请求重定向至 Tomcat 服务器。

使用最为广泛的重定向插件是 isapi\_redirector.dll,同样可以在 apache 组织的官方网站上免费下载,下载之后应选择一个合适的位置保存,通常我们会在 Tomcat 安装目录的 conf 目录下新建一个 connector 来保存和集成相关的文件,因此默认地址为 C:\Program Files\Apache Software Foundation\Tomcat 6.0\conf\connector\isapi\_redirector.dll。

### 3.5 配置 Tomcat 服务器

接下来打开 Tomcat 安装目录的 conf 目录下的配置文件 Server.xml,可以看到五个嵌套元素:Server、Service、Engine、Host 和 Context,其中 Server 级别最高,代表整个 Java 虚拟机,Service 代表虚拟机中的一个服务,Engine 代表服务中的一个请求处理机,Host 表示一个虚拟主机,Context 则表示一个 Web 应用程序。

Service 元素中的 Connector 子元素,表示客户端和该服务之间的连接,具有两个常用属性:Port 属性指定服务器端创建的端口,并在该端口监听从客户端的请求,默认为 8080。EnableLookups 属性指定是否进行 DNS 查询得到远程客户端的实际主机名,默认为 True。

明确了该元素的数据结构,我们可以编辑该文件,将 Service 元素层中的第二个 Connector 元素改为如下内容:

```
<Connector className="org.apache ajp.tomcat6.Ajp13Connector" port="8009" minProcessors="5" maxProcessors="75" acceptCount="10" debug="0"/>
```

经过这样的配置,Tomcat 服务器就会在端口 8009 监听 AJP13 请求,并在接受请求后由类 org.apache ajp.tomcat6.Ajp13Connector 进行处理。我们随后要相应地配置 IIS,使得它用该端口和该协议来发送请求。

### 3.6 配置 IIS

按照惯例,IIS 的重定向配置文件也会放在 Tomcat 安装目录下,所以在 Tomcat 目录的 conf 目录(默认为 C:\Program Files\Apache Software Foundation\Tomcat 6.0\conf\新建两个文本文件,命名为 workers.properties 和 uriworkermap.properties。

workers.properties 文件的正文如下:

```
workers.tomcat_home=C:\Program Files\Apache Software Foundation\Tomcat 6.0(这是默认值,请修改为自己的 Tomcat 安装目录)
workers.java_home= C:\Program Files\Java\jdk1.6.0(这是默认值,请修改为自己的 JDK 安装目录)
worker.list=ServletWorker
worker.ServletWorker.port=8009(默认 8009 端口,和 Tomcat 的配置一致)
worker.ServletWorker.host=local
worker.ServletWorker.type=ajp13(协议名称,和 Tomcat 的配置一致)
```

按照文件的配置,IIS 将会在需要的时机,将来自客户端的请求通过 ajp13 协议,8009 端口,发给名为 ServletWorker 的 worker。

接下来 uriworkermap.properties 文件的正文如下:

```
/examples/*=ServletWorker
/examples/*.jsp=ServletWorker
/examples/servlet/*=ServletWorker
```

其中 example 是假设的 JSP 工程名称,在输入时请修改为自己的工程名称,这三句代码分别表示该工程下所有的静态页面文件,所有的 JSP 页面文件,servlet 文件夹下的所有页面文件均由 ServletWorker 来处理。可以根据不同的需求范围来更改这些代码。

### 3.7 修改注册表

为了让 IIS 配置生效,我们打开注册表 HKEY\_LOCAL\_MACHINE\SOFTWARE,在下面新建目录 Apache Software Foundation\Jakarta Isapi Redirector\1.0,然后添加五个主键:

```
Extension_uri=Jakarta\isapi_redirector.dll(Jakarta 为接下来要创建的 IIS 虚拟目录名)
Log_file= C:\Program Files\Apache Software Foundation\Tomcat 6.0\logs\isapi_redirect.log(请根据安装位置修改)
Log_level=debug
Worker_file= C:\Program Files\Apache Software Foundation\Tomcat 6.0\conf\workers.properties(请根据安装位置修改)
Worker_mount_file= C:\Program Files\Apache Software Foundation\Tomcat 6.0\conf\uriworkermap.properties(请根据安装位置修改)
```

### 3.8 配置 ISAPI 筛选器

这是 Tomcat 和 IIS 集成的最后一步。打开本机的 Internet 服务管理器,用右键点击“默认 Web 站点”,选择“新建”——“虚拟目录”,筛选器的“应用程序名”设置为 Jakarta(必须和 2.7 步的 Extension uri 吻合),“本地路径”设置为 isapi\_redirector.dll 文件所在的目录(这里为 C:\Program Files\Apache Software Foundation\Tomcat 6.0\conf\connector),并且将“执行许可”设置为“脚本和可执行程序”。

新建完毕后,在 Internet 服务管理器主窗口的目录树中,用右键点击 IIS 主机节点,选择“属性”,在出现的属性窗口中,在主属性的“WWW 服务”区域点击“编辑”,增加新的 ISAPI 筛选器,筛选器名称为“Jakarta”,可执行文件为“C:\Program Files\Apache Software Foundation\Tomcat 6.0\conf\connector\isapi\_redirector.dll”。

最后重新启动 IIS,查看 WWW 服务主属性的窗口,如果配置正确,在 ISAPI 筛选器子窗口中,新添加的 Jakarta 筛选器的状态现在应该变为绿色向上的箭头。

### 3.9 集成配置测试

到这里,IIS 和 Tomcat 服务器的集成已经完毕,我们在启动 IIS 和 Tomcat 之后,在浏览器里输入 http://localhost/index.jsp,如果出现和 2.3 步相同 Tomcat 的默认页面,表明这个地址可以不再使用端口 8080,而使用缺省的 IIS 端口 80,再通过 IIS 的筛选器重定向到 Tomcat 服务器,也即集成的效果已经成功。

## 4 结束语

使用 ISAPI 技术集成了 IIS 和 Tomcat 的 Web 服务器,既能够处理原有 IIS 环境下的静态页面请求,又可以接受 Servlet/JSP 机制的请求,充分发挥该机制运行速度快、安全、跨平台等诸多优势,具有很好的实用价值。

### 参考文献:

- [1] 蔡剑,景楠.Java Web 应用开发:J2EE 和 Tomcat[M].2 版.北京:清华大学出版社,2005:43-250.
- [2] 孙鑫.Servlet/JSP 深入详解:基于 Tomcat 的 Web 开发[M].电子工业出版社,2008:2-20.
- [3] Microsoft Corporation.IIS 5.0 使用参考手册——IIS 6.0 新增功能详解[M].北京:科学出版社,2003.
- [4] 李润,李爽,杜瑞庆.JSP 环境配置[J].现代企业教育,2006,11(2):144-145.
- [5] 陈曦,邹桂章.Tomcat 服务器架设技术的探讨[D].广东:广东省计算中心,2003.
- [6] 梁田,倪少权,叶利华.ISAPI 在 Web 服务器中的应用[D].四川:成都信息工程学院学报,2005.