

SIP Servlet 应用路由机制^①

仇 婕^{1,2}, 林 洵², 杨海波^{1,2}

¹(中国科学院研究生院, 北京 100049)

²(中国科学院沈阳计算技术研究所, 沈阳 110171)

摘 要: 3G IMS 系统中, SIP 应用服务器是丰富业务内容的主要提供者之一。为了简化开发流程, 提高开发效率, JCP(Java Community Process)组织定义了 SIP Servlet 标准, 为 SIP 应用服务器开发提供了 Java 应用编程接口。SIP Servlet 应用路由机制是快速定制 3G 业务的重要组件, 成为了当前研究的热点之一。为了自由地组合应用程序完成复杂的业务, 本文在分析 SIP Servlet 默认应用路由的基础上, 提出并实现了利用 SailfinV2 标准容器设计自定义应用路由的方法, 通过一个实例详细阐述了使用该方法的具体步骤, 实验结果表明新增的应用路由机制具有较强的灵活性和高效性。

关键词: SIP Servlet; SailfinV2; 容器; 应用程序组合; 应用路由

Application Router for SIP Servlet

QIU Jie^{1,2}, LIN Hu², YANG Hai-Bo^{1,2}

¹(Graduate University of Chinese Academy of Sciences, Beijing 100049, China)

²(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110171, China)

Abstract: In the system of 3G IMS, SIP application server is the main provider for enriching the content of service. JCP(Java Community Process)defines SIP Servlet Standard to simplify the process and improve the efficiency, which also offers Java application programming interface(API)for SIP application server. The mechanism of application router for SIP Servlet is an important module for rapidly customizing 3G services and it is a hotspot of studying nowadays. In order to implement complex services by composing applications flexibly, this article based on analyzing the application router for SIP Servlet puts forward and realizes a method, which makes use of SailfinV2 standard container to design the custom defined application router, and also expounds how to employ the method according to an instance. Finally, the experiment result reflects the high efficiency and flexibility of newly increased mechanism of application router.

Keywords: SIP Servlet; SailfinV2; container; composition of application; application router

1 引言

3G IMS 系统通过 SIP 应用服务器提供丰富的多媒体业务。为了提高开发效率, 简化开发流程, JCP(Java Community Process)组织定义了建立和部署 SIP 应用程序的容器模型的 SIP Servlet 标准^[1]。容器处理消息格式、信息转发、协议超时等细节信息, 并管理 Servlet 状态信息。基于 SIP Servlet 应用服务器是实现 3G 业务快速定制的重要组件。应用程序开发者只需考虑上层的商业应用服务如何实现, 直接利用 SIP Servlet API(SSAPI)提供相应的业务, 从而简化了工作流程、

降低了开发 SIP 应用服务器的难度, 同时为以后扩展业务提供了技术支持。

SIP Servlet 容器中部署了许多应用程序, 每个应用程序由 Servlets、类文件和其他源程序组成, 各自具有独立的功能。SIP Servlet 规范允许一个请求触发多个应用, SIP Servlet 1.1 标准化了应用程序组合机制, 由两个或者更多应用程序组合完成一个完整的、复杂的服务^[2]。这种模块化的组合方法便于开发者开发新的应用程序, 利于部署者管理功能交互。

SIP Servlet 应用程序组合关键是能正确调用应用

① 收稿时间:2010-04-20;收到修改稿时间:2010-05-31

程序集合提供会话服务。当决定调用哪个应用程序及其顺序时，容器区别对待初始的 Request 请求、随后的 Request 请求和 Response 响应^[1]。因此 SIP Servlet 1.1 标准提出了应用路由机制。

本文通过研究规范中应用路由机制的思想，设计了一个自定义应用路由的实现方案，用来代替默认的应用路由，详细描述了应用路由设计原理与实现的细节。并将此方法应用到一个公司的电话服务系统中，且给出了实验结果及其分析。

2 应用路由基础

2.1 应用路由(Application Router, AR)

JCP 组织先后推出了两个版本的标准规范。在 SIP Servlet 1.0 标准中，应用程序单独的部署在容器中。当容器处理 SIP Request 请求时，依次检查每个应用程序的部署描述符(XML)。若其描述符与 SIP Request 请求相匹配，则转发此请求到目标的 Servlet；否则转发到下一个应用程序。

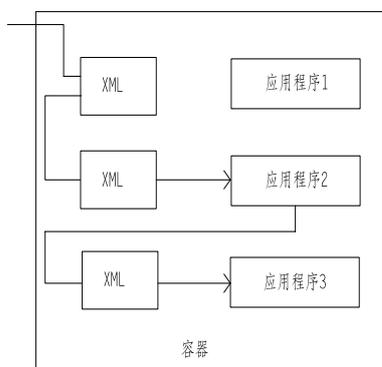


图 1 SIP Servlet1.0 请求路由(Visio 2003)

SIP Servlet 1.1 标准提出了更高效的方法，增加了一个应用路由(Application Router, AR)模块，AR 是负责应用程序选择的特殊组件，可以代理转发 SIP Request 请求给容器。首先容器建立应用程序链，根据当前的状态信息查询 AR，从应用程序链中得到下一步要调用的应用程序名称。当容器收到一个初始的 SIP Request(例如：INVITE 请求建立会话)，查询 AR，如果 AR 返回内部的应用程序，容器转发这个请求至此应用程序；如果 AR 返回了一个外部实体，容器转发

这个请求消息至外部目的地。

应用程序路由没有使用任何应用程序选择算法，仅是一个容器与 AR 之间的 API。SIP Servlet 1.1 标准定义了一个分布式特征组成路由选择机制(DFC-DAR)，其提供了应用程序的优先级和地址，通过控制应用程序的调用顺序，管理组件间的交互。基于一个应用程序选择进程处理初始的 Request 请求，建立一个应用程序路径，随后的 Request 请求和 Response 响应根据此路径执行。

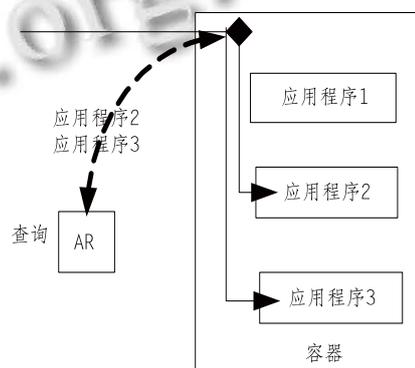


图 2 SIP Servlet1.1 请求路由(Visio 2003)

DFC 思想是把特征看作是独立的组件，所有的消息通过组件从源端传到目的端，组件可以动态的插入到客户与服务器之间的路径上，每个组件不能在不同的连接上共享，但组件相互之间可以进行通信。

2.2 默认的应用路由

SIP Servlet 1.1 标准描述了一种默认的应用路由机制(Default Application Router, DAR)，每行 DAR 配置文件指明了一个或者多个 SIP 方法，随后的 SIP 路由信息以逗号隔开。-Djavax.servlet.sip.ar.dar.configuration 存放着 DAR，存储格式与实例如下^[1]。
INVITE:(sip-router-info-1),(sip-router-info-2)...
SUBSCRIBE:(sip-router-info-3),(sip-router-info-4)...

Sip-router-info 是 SipApplicationRouterInfo 对象的字符串化序列，其包含应用程序名称、DAR 返回的订阅者标识、用 DAR 指令返回 SIP 请求的任何头域(DAR:From)、路由域、路由调节器等。调用 SipApplicationRouter 接口中的 init()方法读取配置文件

信息。

容器与 AR 间的交互主要通过调用 getNextApplication() 方法。当容器收到一个初始 Request 请求, DAR 就调用此方法。getNextApplication() 利用 stateInfo 和初始的 Request 参数找到请求中的 SIP 方法, 之后读取配置文件 sip-router-info, 创建 SipApplicationRouterInfo 对象。

3 自定义AR模型的设计与实现

在 SIP Servlet 容器中可以用自定义 AR 代替 DAR, 其属性定义在 Sipservlet.xml 文件中, 可以任意的修改配置属性。总体设计流程图如下:

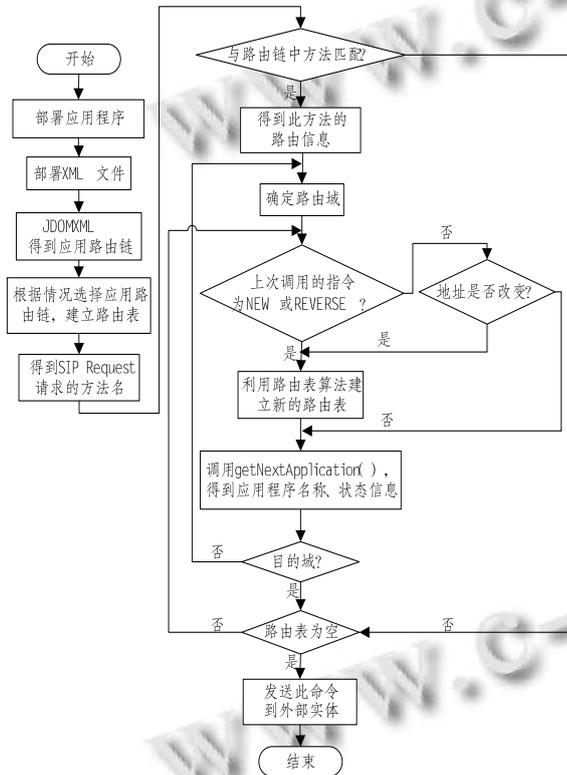


图 3 AR 模块总体流程图(Visio 2003)

3.1 设计配置文件

(1) 路由域

SIP Servlet 容器实例需支持呼叫方和被叫方的应用程序调用, 同时区别不同的订阅者, 据此引入了路由域的概念^[1]。路由域包括原始域、目的域和中间域, 部署者管理路由域信息。当容器查询 AR 调用下一应

用程序时, AR 同时提供路由域信息。应用程序轮流地从容器中获取路由域信息。当容器收到一个初始的 SIP Request 请求时, AR 开始在原始域中选择应用程序。当所有原始域中的应用程序已被调用时, AR 进入中间域并最终进入目的域。

(2) 优先级

对每一个路由域, 配置文件允许部署者建立应用程序的偏序关系^[3]。在 XML 文件 ordering 元素间, 部署者列出应用程序名称。由于, 所部署的一些应用程序与其他程序间没有优先级关系, 因此部署者在配置文件中建立应用程序间的部分序列。根据就近原则应用程序被赋予订阅的优先级。利用特征交互的管理方法, 原始域的应用程序接近呼叫方, 目的域接近被叫方, 容器利用其优先级的高低依次调用各域的应用程序完成业务。

(3) 订阅信息

XML 配置文件中在一个或多个 mapping 元素间描述了 AR 订阅信息^[3]。当确定路由后, 应用路由需在 SipApplicationRouterInfo 对象规定订阅者地址。一个订阅信息由一个或多个 Java 规则表达式和应用程序名称组成。通过提取初始的 SIP Request 请求头域的地址, 确定这个规则表达式的值。原始域的地址是 From 头域的值, 目的域的地址是 To 头域的值。

容器通过订阅信息匹配初始的 SIP Request 请求, 选择应用程序处理此请求。由于可能有多个订阅信息部署在 XML 配置文件中, 且在一个 mapping 元素中, 多个应用程序名字与一个规则表达式相关。因此, 当订阅信息规则表达式匹配初始的 SIP Request 请求地址时, 可能调用多个应用程序处理这个消息。

3.2 定义存储的状态信息

AR 调用应用程序时, 需要存储在容器内的某些状态信息, StateInfo 对象用于存储从当前调用到下一调用的应用路由信息, 它对容器透明, 但是应用程序无权访问。路由信息包括:

- (1) 上次调用的应用程序的地址^[3]。判断此次调用的应用程序地址是否改变, 如果地址改变, 建立新的路由表并根据新路由表信息, 转发 SIP Request 请求。
- (2) 路由表^[3]。它表明处理一个请求需要调用应用程序

的顺序, 作为第一个应用程序是最先被调用处理初始的 SIP Request 请求。当应用程序发出 REVERSR 指令时, 重新建立路由表。每次调用 AR 时, 路由表都要重新计算, 在此情况下, AR 会简单计算整个路由表, 在其中找到上一次调用的应用程序, 并调用路由表中下一个应用程序。

3.3 AR 模块工作流程及实现

当 SIP Servlet 容器收到一个初始的 SIP Request 请求时, 调用 AR 提供给其路由域、路由指令和存储的状态信息。在执行路由选择后, AR 返回给容器下一个调用应用程序的名称、更新的路由域、订阅者地址和更新的状态信息。

3.3.1 AR 模块工作流程

(1) 确定路由域

AR 首先确定当前所在的路由域。如果是初始的 Request 请求, 则处于原始域。当容器发送路由域参数给 AR, 若 AR 确定所有原始域的应用程序都已被调用, AR 进入目的域。

(2) 检查路由指令

确定所在路由域后, 检查上一次调用 SIP Servlet 应用程序的路由指令。路由指令分为: NEW、CONTINUE、REVERSE, 这是一个枚举类型的数据。

当指令为 NEW 时, AR 建立一个新的路由表。如果路由表不为空, 那么把路由表中第一个应用程序的名字返回给容器。反之, AR 进入目的域(此时在原始域)或者返回一个 Null 给容器, 用来表明没有应用程序可以被调用了。

当指令为 CONITINUE 时, 上一个调用的应用程序希望当前域中继续调用应用程序。这种情况下, AR 必须通过对比当前 SIP Request 中的地址与存储在状态信息中的 SIP Request 地址是否一致确定是否需要建立一个新的路由表。如果 SIP Request 中的地址在上一个调用的应用程序中没有改变, 那么先前的路由表仍然有效, 且调用路由表中下一个应用程序; 如果地址改变了, AR 必须建立一个新的路由表。

当指令为 REVERSE 时, AR 必须反转应用程序

链。首先转换路由域, 用建立路由表算法建立新的路由表。REVERSE 指令语义上表明, 调用上一次调用的应用程序之后出现的应用程序, 来处理这个反向的 Request 请求(最后被调用的应用程序是存储在新路由表中的第一个实体)。

3.3.2 建立路由表算法

(1) 根据路由域信息, 从 SIP Request 请求中提取地址。

(2) 在当前的路由域中顺序地检查每个订阅信息, 如果从 SIP Request 提取的地址匹配订阅信息规则的表达式, 则在路由集中增加所有的应用程序名称[3]。

(3) 消除重复。如果任何一个应用程序的名称多次出现在路由集中, 消除重复。

(4) 路由集合排序。根据 XML 文件中的优先级对路由集合进行排序。

3.3.3 实现细节

在用户发出 Request 请求后, 容器调用 AR, AR 调用 init()方法, 此时读取配置文件。利用 JDOM 解析 XML 文件, 输出配置文件的路由信息, 得到一条或者多条应用路由链, 每条路由链都含有所有的 RouterInfo, 根据具体的业务需求选择相应的应用路由链。AR 中定义了一个 HashMap 存储路由相关信息, key 的值为 SIP Request 方法的名称, value 为另一个为存储的 RouterInfo(路由信息)。利用 Request.getMethod()提取 SIP Request 请求的方法名称, 得到此方法的路由信息, 根据建立路由表算法建立应用程序路由表, 调用重写的 getNextApplication()方法, 得到下一个调用的应用程序名称。

(1) 路由信息(RouterInfo)数据结构表如表 1 所示。

表 1 RouterInfo 数据结构表

序号	字段名称	类型	字段说明
1	applicationName	String	应用程序名称
2	subscriberId	String	订阅者的 URI
3	region	String	路由域信息
4	routeUri	String	路由的 URI
5	routeModifier	String	路由调节器
6	stateInfo	Serializable	状态信息

(2) 重写 getNextApplication()方法代码如下:

```

SipApplicationRouterInfo getNextApplication(
SipServletRequest req, SipApplicationRoutingRegion
reg, SipApplicationRoutingDirective dir,
SipTargetedRequestInfo stri, Serializable stateInfo )
{
    checkValidity();/*安全性检查, 是否存在运行时异常*/
    if (req.getURI()!=null) /*接收到一个请求*/
    {
        return createrRouterInfo
        (getNextApplication(req),req,reg)
        /*建立路由信息,得到下一个调用的应用程序名称*/
    };
    else
        return null
    }
}
    
```

(3) 部署自定义 AR

(a) 创建一个类继承 javax.servlet.sip.Sip

Application Router 接口。

(b) 创建一个 Manifest.MF 文件, 此文件中包含 Application-Router 属性: com.path.to.YourRouter[4]。

(c) 把自己编译的代码打包成.jar 并将其存储在 METE-INF 目录下^[4]。

(d) 利用 asadmin deploy yourarchive.java 命令运行。

4 应用举例

本实验中, 选择 SailfinV2 应用服务器和 Netbeans(IDE)平台, 并用 X-lite 终端进行测试, 根据自定义 AR 机制的原理, 设计并实现了一个公司电话系统。

公司电话系统分为三个路径, 每条路径是一条应用路由链:

- 默认路径(default path): 上班时间, 主机正常接听;
- 路径 1(path1): 下班时间, 转接到分机接听;
- 路径 2(Path2): 放假时间, 自动语音留言。

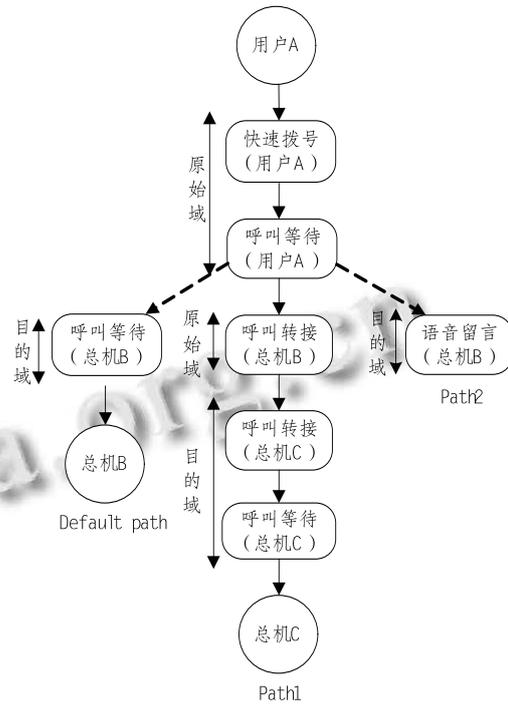


图 4 公司电话服务系统应用程序链(Visio 2003)

(1) 部署配置文件

将三条应用程序路由链信息部署到 SipServlet.xml 配置文件中^[5]。

```

<Orginationg-region>
    <mapping>
        <path id=default>
            <app-name>Speed Dial</app-name>
            <app-name>Call Waiting</app-name>
        </path>
    </mapping>
</Orginationg-region>
    
```

(2) 当用户 A 发送一个 SIP Invite 请求时, 首先根据状态选择路径(以 Path1 为例), 确定选择哪条应用程序路由链处理此 SIP 请求, 利用 JDOM 解析 XML 文件获取应用程序链, 并根据公司状态选择相应的路径。

```

if(RPID.company.work-state==notwork)
    then path 1
if(RPID.company.work-state==holiday)
    
```

```

then path 2
else
    default path

```

(3) 创建 `companyApplicationRouter` 类 implements `javax.servlet.sip.SipApplicationRouter` 接口。

(4) 替换 DAR, 部署自定义 AR。

5 系统测试及分析

为证明增加自定义 AR 提高了通话效率, 在同样环境中, 进行了仿真测试。实验选用 SailfinV2 服务器和 X-Lite 测试终端, 分别对部署与未部署自定义 AR 服务器进行 50 次呼叫, 用 SIPp 测试, 并记录消息响应的的时间间隔, 同时记录计算机内存和硬盘相关参数。实验结果如表 2 所示。

表 2 部署与未部署自定义 AR 的参数比较

SailfinV2 配置	平均时间间隔(ms)	硬盘空间(Mbytes)	内存(Mbytes)
(1) 未部署 AR	151	132	926
(2) 部署 AR	115	137	938
相差值	-36	+5	+12

(1) 未部署 AR

由表 2 可知, 部署了自定义应用程序路由的应用中, 消息的平均时间间隔减少了 23.8%, 而内存和硬盘的占用量仅增加了 3.79%、1.29%。因此, 该机制极大的提高了效率, 且对通话质量影响较小。

6 结束语

作为新兴的技术, SIP Servlet 对于通信业有着重要的意义。为了快速提供用户需要的业务, 本文对 SIP Servlet1.1 标准新增的路由机制进行了研究, 并在此原理的基础上设计并实现一个自定义应用路由(AR), 详述了其数据结构、关键算法和执行过程。最后将此自定义路由应用到公司电话服务系统中。经过长期测试, 系统运行稳定。

SIP Servlet 路由机制增强了业务的灵活性, 提高效率。为了使应用路由能处理动态的、实时的改变, 还需增加一个健壮的路由选择算法, 这将是本文下一步研究的重点。

参考文献

- 1 JSR 289 Java Specification Requests SIP Servlet v1.<http://jcp.org/en/jsr/overview>. 2010,(2):26.
- 2 Cheung E, Purdy KH. Application Composition in the SIP Servlet Environment. IEEE International Conference on Communications, Glasgow. 2007. 1985—1990.
- 3 Cheung E, Purdy KH. An Application Router for SIP Servlet Application Composition. IEEE International Conference on Communications, Beijing, 2008. 1802—1806.
- 4 Sun Microsystems, INC. The SIP Servlet Tutorial, 2009.
- 5 Carlin JME, Thissen D. A SIP Application Router for Presence-Driven Composition of IMS Services. International Conference Next Generation Mobile Applications, Services and Technologies, Gardiff. 2008. 207—212.