

# JAIN - SLEE 中 SIP 资源适配器的设计与实现<sup>①</sup>

## Design and Implementation of SIP Resource Adaptor in JAIN - SLEE

严伟 王纯 (北京邮电大学网络与交换技术国家重点实验室 北京 100876)  
(东信北邮信息技术有限公司 北京 100083)

阮友森 (大唐软件技术股份有限公司 北京 100083)

廖建新 (北京邮电大学网络与交换技术国家重点实验室 北京 100876)  
(东信北邮信息技术有限公司 北京 100083)

**摘要:**简要介绍了业务逻辑执行环境(JAIN - SLEE)中资源适配器连接上层应用和底层事件平台的功能。采用分层结构,设计并实现了会话初始协议(SIP)资源适配器,讨论了其中的一些关键技术。

**关键词:**JAIN - SLEE SIP 资源适配器

### 1 引言

业务逻辑执行环境(JAIN - SLEE)<sup>[1]</sup>是一个容器标准,类似于企业 java Bean(EJB)标准。JIAN - SLEE 容器适用于各种面向消息事件的应用程序,例如多媒体会议、IP 电话服务、分布式交互模拟/监控/控制等。该容器的设计目标就是将移动技术和企业技术整合,能够成为高可用性与可靠性电信级平台。JIAN - SLEE 容器提供一些基本组件,并对运行业务逻辑提供支持和生命周期管理,能够帮助开发人员较快速的开发部署新业务。JAIN - SLEE 主要包括四个部分:业务执逻辑行环境(SLEE)功能、资源适配器框架、外部协议栈和管理部署功能。

资源适配器(Resource Adaptor)是用于具体的协议在 JIAN - SLEE 上的封装,例如 7 号信令系统(SS7)、会话初始协议(SIP)<sup>[2]</sup>、Diameter<sup>[3]</sup>、XML 配置访问协议(XCAP)<sup>[4]</sup>,也可以是自定义的私有协议。资源适配器完成的功能就是将具体协议事件封装,并通知给 SLEE 容器。SIP 协议是下一代网络中重要的信令系统,是国际互联网工程专家组(IETF)多媒体数据和控制体系结构的一部分。SIP 是一个应用层的控制协议,可以用来建立、修改、终止多媒体会话。本文将提供一种设

计和实现 SIP 资源适配器的方案,对 SIP 协议栈的消息事件进行封装。

后续章节安排如下:第 2 节详细描述了 SIP 资源适配器系统结构设计,同时也对各个组成部分进行描述,第 3 节对全文作小结。

### 2 设计与实现

#### 2.1 SIP Resouce Adaptor 总体结构

如图 1 所示,资源适配器在 JIAN - SLEE 容器中的位置是在外部协议栈之上,业务逻辑执行环境之下。其作用是屏蔽外部协议栈的具体实现,将接收到的协议栈的信令消息或特定的事件进行封装,统一上报给业务执行环境。

SIP Resouce Adaptor 是将外部 SIP 协议栈适配到 SLEE 环境之中,将 SIP 消息上报给业务执行环境,并且对 SIP 消息创建和发送进行封装。如图 2 所示,SIP 资源适配器主要是由协议栈适配模块,Activity 管理模块,消息事件通知模块,消息创建模块和消息检查模块组成。

<sup>①</sup> 基金项目:国家杰出青年科学基金(No. 60525110);国家 973 计划项目(No. 2007CB307103);新世纪优秀人才支持计划(No. NCET - 04 - 0111);高等学校博士学科点专项科研基金资助课题(No. 20030013006);电子信息产业发展基金项目(基于 3G 的移动业务应用系统);电子信息产业发展基金重点项目(下一代网络核心业务平台)

## 2.2 协议栈适配模块设计

协议栈适配模块主要负责将外部的协议栈屏蔽,从 SLEE 角度来看可以将协议栈作为一个黑盒。

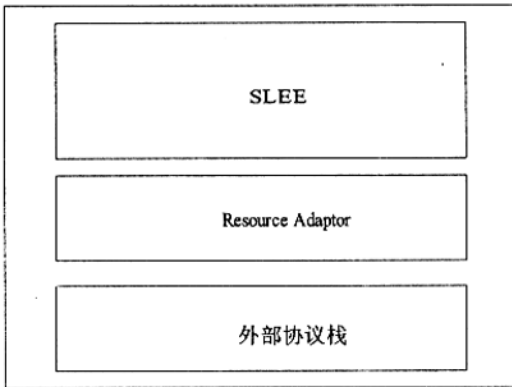


图 1 整体框架

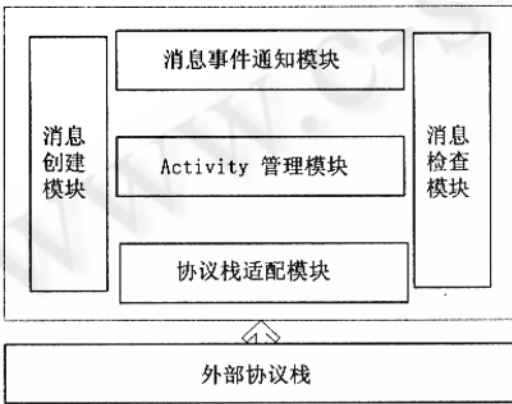


图 2 SIP 资源适配器的模块结构

对于 SIP 资源适配器,协议栈适配模块根据 SIP 协议栈的实现不同,可能存在两种设计方式:

(1) SIP 协议栈提供了观察者模式 (OBSERVER)<sup>[5]</sup> 来使客户端接收 SIP 消息,则协议栈适配模块可以作为 SIP 协议栈的观察者,负责接收协议栈收到的 SIP 消息。

(2) 协议栈适配模块直接从网络上接收 SIP 消息,这就需要该模块能够具有较高性能处理网络数据接收的能力。该模块设计可以参考图 3,底层的 Socket 接收发送可以使用 Java™2 Platform Std 提供的 java. nio 包,该包专门提供了一个面向块的 I/O 系统以块的形式处理数据,并采用了非堵塞技术,可以大大提高 I/O 读取效率。该设计中采用了线程池技术,主要是考虑到突发的网络消息到来的速度很快,如果对每个网络消息都创建单独的线程去处理,可能开销太大;如果不

使用单独线程去处理网络消息,整个消息处理都会处于阻塞的状态,当一个消息未处理完成无法处理新消息。综合考虑之下,采取了线程池方式来处理。系统启动时,在线程池内分配好实例,当从网络中接收到消息后,就将该消息委托给线程池的实例去处理,这样既可以减少频繁创建线程的开销,又可以提高网络数据的处理速度。消息处理模块就是将 SIP 消息进行编解码操作以及对 SIP 中的事务进行管理,并将消息提交给上层模块处理。

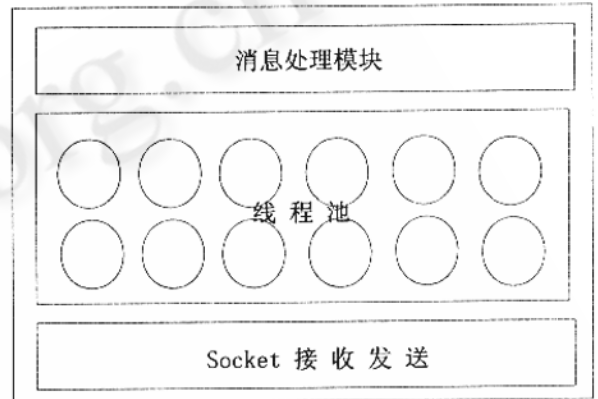


图 3 协议栈适配模块

## 2.3 Activity 管理模块设计

Activity 是资源适配器中的重要概念,对于 SIP 消息来说,可以理解为 SIP 中的事务。Activity 的管理主要是对其在资源适配模块中生命周期进行管理。图 4 显示了 Activity 的生命周期。

Activity 保存的实现可以使用 java 中的 java. util. Map 类型。

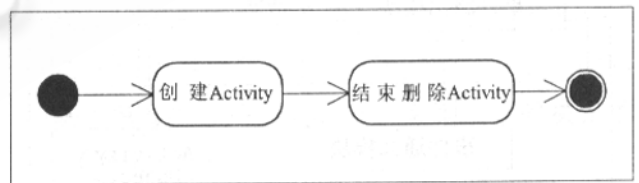


图 4 Activity 生命周期

Activity 的创建过程:

- 消息处理模块收到 SIP 消息时,将创建新的 ServerTransaction,并将事务中的 branchID 作为主键、Transaction 作为值加入到 Activity Map 中,用来记录相应的呼叫状态。

· 消息处理模块收到 SIP 响应时,将创建新的 ClientTransaction,并将此事务中的 branchID 作为主键、Transaction 值加入到 Activity Map 中。

Activity 的回收过程:

· Activity 的回收是通过两个扫描的线程来完成的。第一个线程扫描 Activity Map 中的 Transaction 的状态,如果状态为 TERMINATED,则代表该事务已经结束,则将其放入 TerminatedActivity Map 中,并启动第二个线程。第二个线程则进行扫描 TerminatedActivity Map,如果存在相应的键,向 SLEE 通知该事务已经结束,并删除该键和值。

· 如果协议栈提供事务结束事件的异步通知机制,则当一个事务结束后,协议栈将此事件通知给消息处理模块,将在 Activity Map 中删除已经结束的事务。

### 2.4 消息事件通知模块设计

如图 5 所示,事件通知模块是业务执行环境和资源适配器之间的连接点。所有的资源适配器需要上报给 SLEE 的消息,都是通过该模块提交给 SLEE 中的事件路由模块所处理,并由事件路由模块将消息传递到部署在 SLEE 中的业务构建块(SBB)。该模块设计需要考虑能够保证 SIP 消息按照顺序提交给 SLEE,并且能够保证业务发出的 SIP 请求的消息的响应顺序。业务使用 SIP 资源适配器中的消息创建模块创建新的 SIP 请求,消息事件通知模块必须能够保证该请求的响应顺序接收。

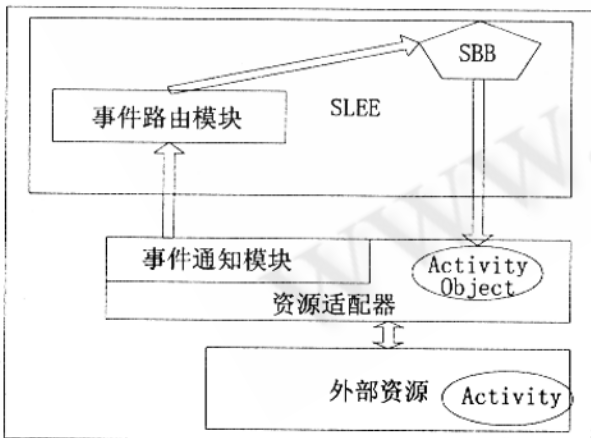


图 5 消息事件上报模型

### 2.5 消息创建和消息检查模块设计

消息创建模块主要是使用抽象工厂模式 (AB-

STRACT FACTORY)<sup>[5]</sup>或工厂方法模式 (FACTORY METHOD)<sup>[5]</sup>来封装协议栈提供的创建消息的方法,业务执行环境中的业务能够较方便使用工厂创建消息。

消息检查模块主要是负责对请求的消息进行认证。对于 SIP 消息,可以使用 Digest<sup>[6]</sup>方式,也可以使用 3GPP IMS 所采用的认证和密钥协商 (AKA) 方式。该模块作用是拦截那些没有通过认证的消息,不将这些消息上报给业务执行环境, SIP 资源适配器直接对这些消息发送响应。

### 3 总结

业务逻辑执行环境将会广泛应用于下一代网络中,对在下一代网络中提供各种增值业务具有非常关键的作用。因此,研究、设计和实现一个高性能的资源适配器是很重要的。本文描述了 JAIN - SLEE 环境中 SIP 资源适配器设计和实现方法,并可以参照此方法设计和实现其他协议的适配器。经实践证明,使用本文设计的 SIP 资源适配器运行稳定,并具有良好的处理性能。

### 参考文献

- 1 Swee Boon Lim, David Ferry. JAIN SLEE 1.0 Specification, Final Release, 2004.
- 2 J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler. SIP: Session Initiation Protocol. RFC3261, IETF, 2002.
- 3 P. Calhoun, J. Loughney, E. Guttman, G. Zorn, J. Arkko. Diameter Base Protocol. RFC3588, IETF, 2003.
- 4 J. Rosenberg. The Extensible Markup Language (XML) Configuration Access Protocol (XCAP). draft - ieff - simple - xcap - 10, IETF, 2006.
- 5 Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Design Patterns: Elements of Reusable Object - Oriented software. Pearson Education, 1994.
- 6 J. Franks, P. Hallam - Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart. HTTP Authentication: Basic and Digest Access Authentication. RFC2617, 1999.