

# 企业服务总线路由器的研究与设计<sup>①</sup>

刘小艳<sup>1,2</sup>, 廉东本<sup>2</sup>

<sup>1</sup>(中国科学院研究生院, 北京 100049)

<sup>2</sup>(中国科学院 沈阳计算技术研究所, 沈阳 110171)

**摘要:** 企业服务总线是一种基于 SOA 架构, 能够实现应用集成的软总线. 通过分析已有 ESB 路由的缺陷, 提出改进后的跨 ESB 的路由设计方案. 它实现了基于服务规则和路由策略的动态消息路由, 并且能够跨 ESB 进行交互. 实际运行系统表明该设计方案有良好的应用效果.

**关键词:** ESB; 企业服务总线; 路由器; 消息路由; 动态路由

## Research and Design of Router of Enterprise Service Bus

LIU Xiao-Yan<sup>1,2</sup>, LIAN Dong-Ben<sup>2</sup>

<sup>1</sup>(Graduate School, Chinese Academy of Sciences, Beijing 100049, China)

<sup>2</sup>(Shenyang Institute of Computing Technology of Chinese Academy of Sciences, Shenyang 110171, China)

**Abstract:** Enterprise Service Bus is a kind of software bus based on SOA architecture. It can achieve application integration. We improved the ESB-across router by analyzing the defects of the existing ESB routing. It implements the dynamic message routing based on rules and routing strategies, and can across the ESB to interact. The actual operation of the system shows that the design has good effect.

**Key words:** ESB; enterprise service bus; router; message routing; dynamic routing

企业服务总线<sup>[1]</sup>(Enterprise Service Bus, ESB)是 SOA 架构中实施应用集成的一种方式, 是传统中间件技术与 XML、Web 服务等技术结合的产物. 第三方的应用程序能够以标准的方式“插入”到该平台上运行, 各应用程序只需遵循 ESB 所支持的消息格式, 由企业服务总线完成服务请求者与提供者之间具体的消息格式转换以及路由选择, 实现了各个服务的访问位置和传输协议的透明化. ESB 凭借其动态且可靠的消息转换和传输能力, 有效地解决了应用集成中普遍存在的通信协议不同以及编程语言差异等异构问题, 具有很强的灵活性和可扩展性, 应用前景良好.

ESB 路由器是企业服务总线的核心部分. 我们将针对现存 ESB 路由的设计缺陷<sup>[2-4]</sup>提出改进. 第一, 现今大部分 ESB 路由只能支持静态路由, 即使用静态的配置文件来定义路由. 而在现实应用中, 我们不能避免同一 ESB 中存在同种服务, 这就提出了动态选择服

务提供者的需求. 据此我们设计了基于服务规则和路由策略的动态消息路由方案. 该方案中, 服务规则库和路由策略都将作为动态路由选择的依据. 第二, 为了提高 ESB 的应用集成能力, 我们将设计跨 ESB 的路由方案.

跨 ESB 的路由器将服务注册到不同的 ESB 服务器上, 通过 ESB 间的交互实现不同 ESB 结点上不同服务间的交互. 我们将 ESB 路由器的功能分成两块: ESB 之间路由和 ESB 内部路由. 下面我们将主要从这两方面来设计 ESB 路由器.

### 1 ESB之间路由的设计

ESB 之间路由就是将服务请求路由到服务所挂接的 ESB 结点. 由于服务只允许它所挂接的 ESB 节点调用(处于安全性等的考虑), 因此要访问其他 ESB 上注册的服务必须进行 ESB 间的路由. 我们通过 ESB

① 基金项目: 国家水体污染控制与治理科技重大专项(2012ZX07505003)

收稿时间: 2012-07-20; 收到修改稿时间: 2012-08-23

之间建立信任关系来建立 ESB 之间的信息通道. 之后我们将生成路由路径并进行消息路由<sup>[5]</sup>. 下面详细介绍 ESB 间信任关系的建立和路由路径的生成. 图 1 描述了服务注册关系和 ESB 间的信任关系.

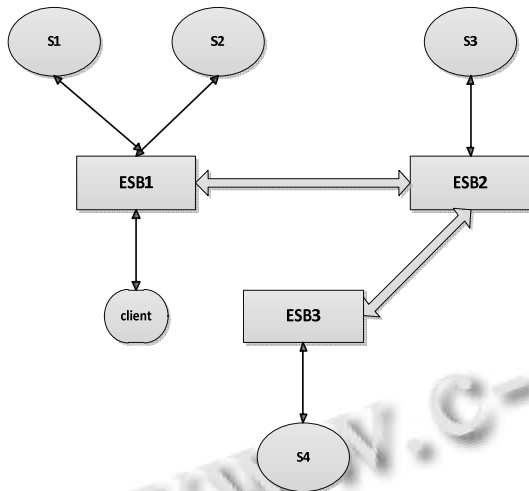


图 1 ESB 间信任关系示意图

### 1.1 建立 ESB 之间信任关系

当两个 ESB 之间(假设为 ESB1 和 ESB2) 提出交互需求时, 那么两者要建立信任关系. ESB1 请求与 ESB2 建立建立信任关系的过程为:

- ① ESB1 先获取 ESB2 给予的认证码.
- ② ESB1 发送“请求建立互信消息”(包括认证码、ESB1 上注册的服务信息等) 给 ESB2.
- ③ ESB2 收到消息后, 获取消息中的【认证码】验证消息的可靠性. 认证成功后, 将 ESB1 发送来的服务信息存储到 ESB2 的服务注册表中, 并标记为 exchange.
- ④ ESB2 将 ESB2 上注册的服务信息发送给 ESB1.
- ⑤ ESB1 将收到的服务消息存储到服务注册表, 并标记为 exchange.

信任关系的建立关键是服务信息的交换. 在交换服务信息时, 只交换 ESB 下注册的服务信息, 而通过交换获取的信息(即标记为 exchange 的服务信息)不参与交换. 按照图 1 所示建立信任关系的结果如表 1 所示.

表 1 服务信息表

	ESB1		ESB2		ESB3	
	name	type	name	type	name	type
			e			

建立信任关系之前	ESB1	center	ESB 2	center	ESB3	center
	S1	local	S3	local	S4	local
	S2	local				
ESB1 与 ESB2 建立信任关系后新增	ESB2	Ex-center	ESB 1	Ex-center		
	S3	Ex-local	S1	Ex-local		
			S2	Ex-local		
ESB2 与 ESB3 建立信任关系后新增			ESB 3	Ex-center	ESB2	Ex-center
			S4	Ex-local	S3	Ex-local

### 1.2 生成消息路由路径

跨 ESB 请求服务时, 从服务信息表中寻找源结点与目的结点之间的路径, 即源结点和目的节点是否是信任关系. 如果存在可达路径, 则源结点和目的节点是信任关系, 两者可以通信; 否则, 不能通信. 生成消息路由路径的算法如下.

ESB1 收到服务请求后, 查询服务信息表, 进行如下操作:

- ① 如果目标服务是本地服务(local), 返回 ESB1.
- ② 如果目标服务是从 ESB2 交换来的服务(exchange local), 返回 ESB1+ESB2.
- ③ 如果不是 1 或 2, 则将消息转发给所有直接信任结点. 收到消息的 ESB 节点也按照以上步骤进行路径生成过程.

这是一个递归的过程, 最终生成消息路由路径. 之后按照路径依次转发消息到最终目的 ESB 节点, 再由它调用目标服务.

## 2 ESB 内部动态消息路由的设计

ESB 路由器结构图如图 2 所示.

ESB 路由器收到 ESB 内部标准消息后, 由消息引擎对它进行解析, 并结合路由策略将其放入消息缓冲器中; 消息缓冲器负责所有请求消息的缓冲管理, 根据不同的路由策略消息缓冲器维护不同的消息队列. 之后根据上节所述生成消息路由路径. 检查目的节点是不是自己, 如果不是, 则进入消息转发模块, 将消

息转发到其他 ESB 服务器, 否则进入 ESB 内部路由。

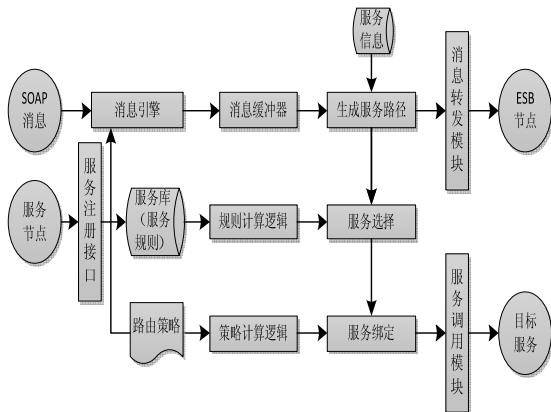


图 2 ESB 路由器结构图

ESB 内部路由: 如果所请求的服务在本地, 则进入服务寻址单元, 根据服务规则选择符合要求的服务集<sup>[6]</sup>。然后根据路由策略从合法服务集中选择最佳服务进行服务绑定, 即为服务调用模块提供服务地址。ESB 路由器的动态性主要体现在两个方面:

- ① 服务及服务规则的动态添加、更新及删除, 实现动态服务寻址。
- ② 存在多个同名服务时, 依据动态路由策略实现动态路由绑定。

下面从这两个方面设计动态消息路由。

### 2.1 基于服务规则的动态消息路由

数据在进行服务选择时, 要依据各个服务的规则要求进行筛选。服务规则是指访问服务所要遵循的条件, 比如对服务请求者的地域限制、对服务请求者的类型限制等。

考虑到动态调整服务对象规则的需求, 我们的服务规则将由各个服务来控制。这样提高了服务规则更新的灵活性。为了动态更新服务规则, 我们把服务注册接口作为服务节点和服务规则库之间的消息通道, 以便服务节点把自己所提供服务的规则变更情况及时反映到 ESB 服务规则库中。

服务注册接口程序是由 ESB 路由器提供的, 除了支持服务规则的添加、删除和修改, 该接口主要提供服务注册、服务更新和服务删除等功能以实现服务注册模块。服务提供者只需调用 ESB 的服务注册接口, 就可以实现服务的动态更新, 为动态服务选择提供了坚实的基础。

可能的服务注册接口如下:

```
class ServicesRegister
```

```
{
    //1.创建服务
    CreateService( Service s );
    //2.更新服务,包括服务名、服务路径等的修改
    UpdateService( Service s );
    //3.根据服务 ID 删除对应服务信息
    RemoveService( String ServiceID );
    //4.根据服务 ID 创建新的服务规则
    CreateServiceRule(String ServiceID, String
        ruleName, String ruleValue);
    //5.根据服务 ID 和规则名称更新规则的值
    UpdateServiceRule(String ServiceID, String
        ruleName, String ruleValue);
    //6.根据服务 ID 和规则名称删除规则
    RemoveServiceRule(String ServiceID, String
        ruleName);
    .....
}
```

### 2.2 基于路由策略的动态消息路由

ESB 系统中, 同种服务往往存在多份, 因为这种设计具有很大的现实意义。首先, 对于一个大型分布式系统来说, 必然存在很多服务提供者, 这就导致出现重复的服务。由此服务提供者之间存在竞争关系, 能促使服务提供者不断改进自己的服务。另外, 服务资源一定的冗余度, 可以确保在某些服务不可用的情况下, 系统能够自动路由到其他服务提供者, 从而保证系统能够继续正常运行。这样极大地提高了系统的健壮性和可用性。

为了解决同种服务存在多份的问题, 我们设计了路由器的服务绑定模块。该模块从备选列表选择一个服务提供者, 并给出其地址, 而选择的依据就是路由策略。常见的路由策略有: 负载均衡策略, 容错策略, 最小响应时间策略, 费用最小策略等等。这些策略可以单独使用, 也可以组合使用。比如选择时先考虑负载均衡, 再考虑最小响应时间等。

服务总线路由器实现了基于负载均衡的规则和基于容错的规则, 用户在具体使用时也可以根据系统的功能需求自定义新的路由规则完成消息的路由。

ESB 路由器需要一张路由表辅助工作。ESB 系统启动时, 内核读取服务库生成路由表, 并在系统运行

中, 动态更新内容以即时反应服务信息的变化. 路由表维护了服务 ID 与服务地址的映射关系, 在服务绑定结束后, 服务调用模块根据筛选出的服务 ID 从路由表中查找对应的服务提供者的地址和服务的参数信息进行服务调用. 另外, 为了给路由策略计算逻辑提供依据, 路由表还记载了服务费用、服务的最大并发请求数、处理中的请求数、上次调用的响应时间、平均响应时间等信息.

可能的路由表如表 2 所示.

表 2 路由表

字段名称	数据类型	长度	能否为空	字段说明
ServiceID (PrimaryKey)	int	20	N	服务 ID 号(主键)
ServiceName	string	40	N	服务名称
URL	string	100	N	服务地址
InputParams	string	100		输入参数
OutputParams	string	100		输出参数
ServicePrice	double	20		服务价格
MaxRequest	int	20		最大并发请求数
Finishing Request	int	20		正在处理请求数
LastResponseTime	int	20		上次调用响应时间
AvgResponse Time	Int	20		平均响应时间

(上接第 160 页)

术主要是以供应链管理为平台, 以综合物联网感知技术与最新监控技术为手段, 以互联网为纽带的多种通讯方式来实现运输的全方位管理, 是未来运输的必然趋势.

### 参考文献

- 1 苏彬, 范曲立, 宗平. 物联网的体系结构与相关技术研究. 南京邮电大学学报(自然科学版), 2009(29): 1-11.
- 2 钟乐海. 多功能网络实时监控系统的实现. 计算机应用,

### 3 结语

本文所设计的企业服务总线路由器改变了传统 ESB 静态路由机制, 设计了基于动态服务规则和路由策略的动态消息路由方案. 同时设计跨 ESB 的路由提高系统应用集成能力. 该 ESB 路由器的设计已经应用到实际项目中, 取得了较好的效果. 今后还要对路由算法及路由策略的使用进行进一步的研究和实现.

### 参考文献

- 1 曾文英, 赵跃龙, 齐德昱. ESB 原理架构实现及应用. 计算机工程与应用, 2008, 44(25): 225-228.
- 2 符宁, 周兴社, 杨刚, 等. 分布式企业服务总线的设计与实现. 计算机工程, 2007, 33(24): 118-120.
- 3 李秀林, 张祖平. 基于动态消息路由的 ESB 框架的研究与应用. 计算机系统应用, 2009, 18(9): 132-135.
- 4 符宁, 周兴社, 张海辉. 基于 JMS 的分布式 ESB 的设计与实现. 计算机科学, 2007, 34(12): 118-121.
- 5 原鑫, 毕会娟, 张立松, 柴媛媛. 面向服务的消息中间件研究. 微计算机信息, 2008, 24(10): 250-251.
- 6 张晖, 李国栋, 柳长安. 基于接收表的 ESB 研究与设计. 电子科技, 2009, 22(11): 47-49.

2003, 1: 126-128.

- 3 李凤保, 李凌. 无线传感器网络技术综述. 仪器仪表学报, 2005, 26(8): 77-80.
- 4 International Telecommunication Union UIT. ITU Internet Reports 2005: The Internet of Things, 2005.
- 5 曹青林. 物联网研究现状综述. 软件导刊, 2010, (5): 6-7.
- 6 吴洲, 等. GPS 车辆监控系统在企业物流的应用. 计算机与数字工程, 2007, 35(9): 164-166.