

# 基于内容的 ESB 消息路由机制<sup>①</sup>

吴高峰<sup>1,2</sup>, 丁君辉<sup>1</sup>, 徐远兵<sup>1</sup>

<sup>1</sup>(湖南中烟工业有限责任公司 长沙 410014)

<sup>2</sup>(中南大学 信息科学与工程学院, 长沙 410014)

**摘要:** 企业服务总线(ESB)能够将基于面向服务的体系结构(SOA)架构的各种企业应用系统进行整合, 消息路由是 ESB 的核心功能, 现有的 ESB 路由机制都是静态的, 依赖于繁琐的配置文件, 限制了路由的灵活性, 同时, 随着服务数量的增加, 配置服务信息呈指数级增长, 影响系统性能. 针对这些问题, 首先提出了一种基于内容的消息路由机制, 然后对消息格式、路由表、工作流程和算法实现进行了详细描述, 最后, 通过实验, 从功能和性能两方面验证了所提方法可行性.

**关键词:** ESB; SOA; 基于内容的路由

## Content-Based Routing Based on ESB

WU Gao-Feng<sup>1,2</sup>, DING Jun-Hui<sup>1</sup>, XU Yuan-Bing<sup>1</sup>

<sup>1</sup>(China Tobacco Hunan Industrial Co. Ltd., Changsha 410014, China)

<sup>2</sup>(School of Information Science & Engineering, University of Central South of China, Changsha 410014, China)

**Abstract:** Enterprise Service Bus (ESB) provides the integration of multi application systems in Service-Oriented Architecture (SOA). Message routing is the core and basis of ESB. At present, the routing mechanisms in ESB software are all static, relying on tedious configuration file, limited routing flexibility. At the same time, along with the increase of the number of services, service configuration information is growing exponentially, affect system performance. In order to solve these problem, a method of content-based routing is presented firstly. Then we introduced the messages format, route table, work process and the algorithm in detail. Finally, experimental results showed that the proposed method was efficient and reliable.

**Key words:** ESB; SOA; content-based routing

随着信息化的快速发展, 在企业内部建立了许多功能各异、相对独立的应用子系统, 如生产指挥调服系统、营销综合管理系统、财务管理系统等. 这些系统无论是从业务和功能方面, 还是从系统架构、开发环境、数据库等方面都存在着差异, 随着时间的推移和技术的进步, 这些相互隔离的信息服务和管理系统形成了一个“信息孤岛”, 而不断变化的业务需求往往需要将建于不同时期、不同类型的异构系统之间的数据、流程、服务进行集成和重组, 形成资源整合和信息共享, 以便为用户提供简洁高效的信息服

务. 企业服务总线(Enterprise Service Bus, ESB)作为面

向服务的架构(Service-Oriented Architecture, SOA)思想的最佳实现, 它将中间件和 XML、Web Services 等技术整合起来, 提供了基于标准的、松散耦合的、灵活性和扩展性非常高的消息机制, 通过适配器实现与其他组件互连互通, 能很好的解决上述的企业应用集成问题. 目前, 在 ESB 领域不仅存在着以各大软件厂商主导的商业 ESB 产品, 而且还活跃着许多开源 ESB 产品, 前者以 IBM 的 WebSphere 和 BEA 的 AquaLogic Service Bus 为典型代表, 后者主要有 Mule、Celtix 和 ServiceMix 等. 这些 ESB 产品实现的基本功能有: 消息路由、消息通信(寻址、协议和标准、发布/订阅、同

① 基金项目: 湖南中烟科技支撑计划(KY2011XX0002)

收稿时间: 2014-04-23; 收到修改稿时间: 2014-06-06

步和异步消息传递等)、服务交互、集成、服务质量、消息处理(消息内容、协议的格式转换)和服务监管和安全。

其中消息路由是 ESB 的核心功能,通过消息路由,服务请求方在发送服务请求后无需关心是哪个服务提供者接收并响应它的请求,实现了服务消费者和服务提供者间的成功解耦。目前,无论商业 ESB 产品还是开源 ESB 大多采用的是固定路由,这是一种点对点的路由方式,通常是在请求消息头文件中硬性书写路由规则,基于消息的入口点来确定消息的出口点,限制了路由的灵活性。在性能方面,由于 n 个服务的互相通信需要配置 O(n)级的通道,随着服务数量的增加,配置文件信息呈指数级增加,造成性能的下降。因此,本文提出基于内容的路由(Content-based Routing, CBR)算法,降低了系统的成本而且配置灵活,提高了系统的实时响应性。

### 1 基于内容的消息路由实现

基于内容的路由首先解析消息的内容,通过事先配置的路由规则和逻辑,由消息内容决定目的地址,能将每一条输出的消息直接发送给正确的应用系统服务。如图 1 所示。

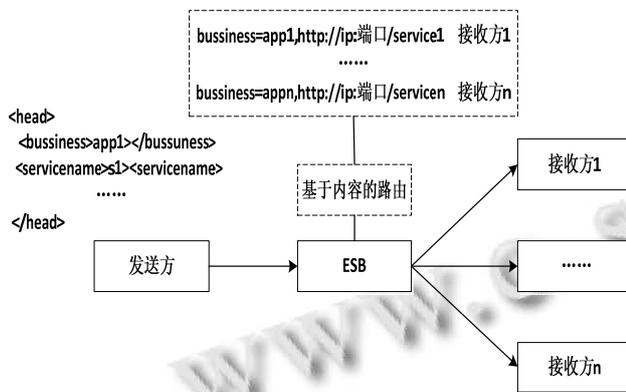


图 1 CBR 路由实现

我们重点描述 CBR 消息格式、路由表构造、路由算法的实现及实现流程。

#### 1.1 CBR 消息格式

ESB 中的消息是基于 XML 格式的,当服务请求者以 XML 格式流入 ESB 后,ESB 中的消息代理侦听到该消息,将 ESB 交给 ESB 其他组件,这些组件分析处理后,再将消息发送给正确的目标服务提供者。结合本文所提的 CBR 算法,设计了服务请求消息。服务请

求消息主要分为两部分:消息头(head)和消息体(body)。

消息头记录消息属性,用于控制消息的传递,主要包括 1)business,业务流程的名称,对各业务系统进行惟一标识; 2)servicename,调用服务的名称; 3)sender,记录消息的来源; 4)status,记录消息传递的状态。

消息体记录传输的实际内容,主要包括: 1)id,所有消息中能唯一标识此消息的主键; 2) category,消息类别; 3)from,应用系统名称; 4) content, 消息内容

服务请求消息格式如下所示:

```

<root>
  <head>
    <bussiness> </bussiness>
    <servicename> </servicename>
    <sender> </sender>
    <status> </status>
  </head>
  <body>
    <id> </id>
    <category> </category>
    <from></from>
    <content> </content>
    .....
  </body>
</root>

```

#### 1.1 CBR 路由表

路由表是一组规则的集合,是以 XML 格式表示的一个可配置文件,可以用二元组描述 route:=<bussiness, {service}>,其中 bussiness 表示业务流程的名称, {service}表示提供服务的集合,可以进一步细化为 {service}:=<servicename,url>, servicename 表示提供服务的名称, url 表示服务的地址,如图 2 所示。

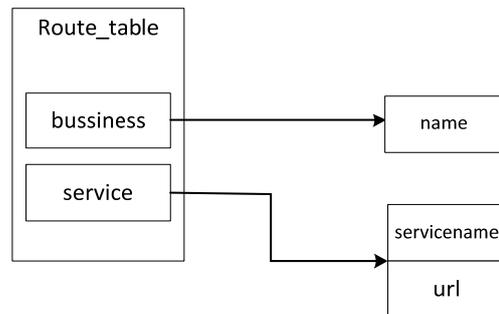


图 2 路由表结构

路由表的示例结构如下:

```

<root>
<R1>
<business>
<service1>servicename=service1,url=""</service1>
<service2>servicename=service2,url=""</service2>
.....more service.....
</business>
</R1>
<R2>
<business>
<service1>servicename=service1,url=""</service1>
<service2>servicename=service2,url=""</service2>
.....more service.....
</business>
</R2>
.....Ri.....
</root>
    
```

1.2 CBR 路由算法

根据图 1, 可以将 CBR 抽象为一个四元组, <Ms, Rule, Router, Reciver>, Ms 为服务调用方发出的 XML 消息, Ms.head 为 Ms 头部属性; Rule 表示路由表规则集合, Rule = {R1, R2, ....., Ri}, R 为服务单元 i 的规则集, i ∈ N, 相关的操作是 add(rule)和 remove(rule); Router 表示对基于内容路由的相关操作, 包括规则匹配和转发服务请求消息 sender(message,r),rReciver; Reciver 表示多个服务提供者, Reciver=(r1, r2, ....., ri), i ∈ N, 相关的操作是 reciver(message).

核心代码如下:

```

(1)Ms=(TextMessage)getMessageFromQueue();
(2)XmlDocument doc = new XmlDocument();
(3)Doc.loadXml((String) Ms);
(4)XmlElement root=doc.DocumnetElement;
(5)business ← root.selectSingleNode("head/
business");
(6)servicename ← root.selectSingleNode
("head/ servicename");
(7)for Ri ∈ R do {
(8) if (Ri.business ==business
and Ri.servicei==servicename)
    
```

```

Service destination = BFS(servicename,url);
destination.sender (Ms, r);
}
    
```

其中,(1)读取服务调用方发来的消息, (2),(3) (4),(5)解析消息头部信息, 得到业务流程名称和服务名称,(7),(8)进行路由匹配并转发消息.

1.3 CBR 流程

CBR 流程描述如下, 流程图如图 3 所示.

Step1: 服务调用方消息到达 ESB 后, ESB 接收请求消息, 然后调用访问控制模块, 判断该服务请求是否有权访问服务;

Step2: 如果无权限访问, 则调用异常处理模块进行异常处理, 然后将异常响应消息通过发送响应模块发送给服务请求者, 并记录日志, 否则转 3;

Step3: 通过内容解析器来解析消息的头部属性, 提取业务流程和服务名称, 并以此为路由匹配条件去查找路由表;

Step4: 当找到所调用的服务后, 通过服务调用模块调用服务提供方提供的服务;

Step5: 当服务调用完后, 通过响应消息转换模块, 把服务提供方返回的响应消息转换为服务请求方需要的响应消息格式, 然后通过发送响应消息模块, 把响应消息返回给服务调用方.

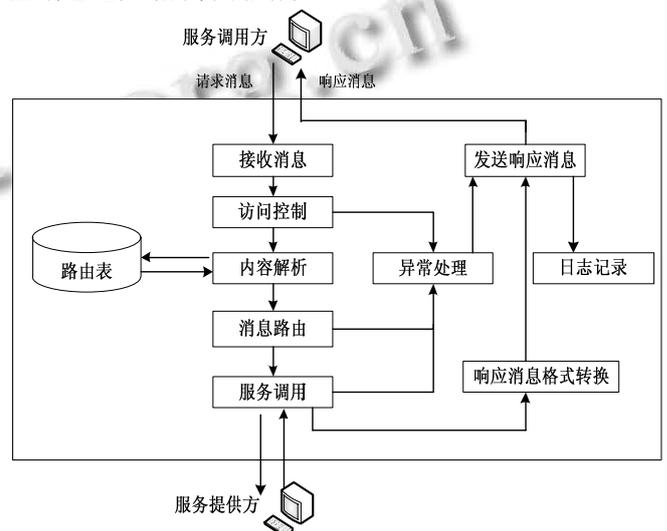


图 3 ESB 工作流程

2 实验方法及结果分析

为了评估本文提出的 CBR 算法, 描述了以下二个

实验场景: 服务响应时间和 CBR 并发性能. 实验环境如下: 客户端包括 Pentium Dual-Core 2.8G/4GRAM 的配置及用来预测系统行为和性能的负载测试工具 loadRunner; 基于 CBR 的 ESB 中间件及商业的 ESB 产品 (CBR-ESB), IBM WebSphere Message Broker v6 (MB-ESB); 服务器端为 Pentium Dual-Core 2.8G/4GRAM 的多台主机, 用来提供服务. 实验部署图如图 4.

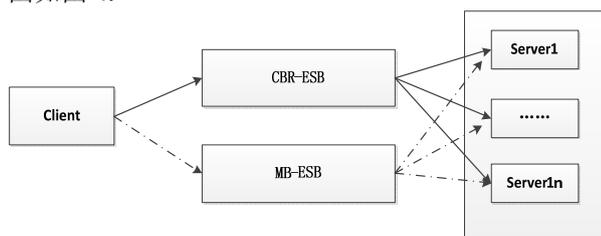


图 4 实验部署图

### 2.1 服务响应时间

表 1 显示的是服务调用方通过 ESB 总线与服务提供方进行数据交换时的平均响应时间, 包括数据发送和返回两部分时间. 在请求次数较少时, 两者的响应时间差别不大, 当请求次数大于 1000 时, 采用静态路由的 MB-ESB, 由于配置文件的增大, 与采用基于内容的路由相比, 需要更长的响应时间.

表 1 服务响应时间

请求次数 \ 平均响应时间	50	1000	5000	10000
路由方式				
MB-ESB	12ms	150ms	9.6s	32s
CBR-ESB	11ms	120ms	0.4s	1.8s

### 2.2 并发测试

测试系统的并发性能, 通过客户端程序创建若干请求, 并发送到 ESB 总线, ESB 将请求转化为 XML 格式后, 分发到目标服务系统, 统计客户端通过总线与目标服务系统进行数据交换的平均响应时间. 与 IBM WebSphere Message Broker v6 (MB) 相比, 从图 5 可以看出, 本文实现的 ESB 在平均响应时间方面优于 MB. 随着并发数的增加, 本文由于采用了基于内容的路由算法, 响应时间呈线性增长, 而 MB 随着并发数的增加, 静态路由配置文件加大, 载入内存时间变长, 变化幅度较大.

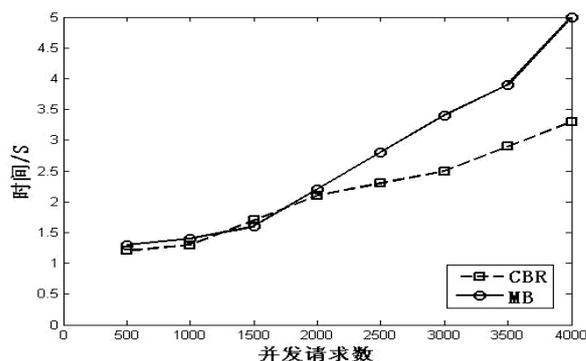


图 5 ESB 并发性能测试

### 3 结语

本文提出了基于内容的路由数据交换总线, 通过使用该 ESB, 总体来说, 可以以无缝的非侵入方式使湖南中烟已有系统具有全新的服务接口, 支持新系统的快速接入和已有系统简约变更, 降低了系统的耦合性, 提高了系统的适应性和扩展能力, 满足了用户数据交换需求.

### 参考文献

- 1 陈传波, 徐峰, 陈世雄. 基于 XML 消息的安全研究. 计算机工程与科学, 2004, 26(8): 1-3.
- 2 Schmidt Mt, Hulchimm B, Lambrm P, et al. The enterprise service bus: Making service-oriented architecture real. IBM System Journal, 2005, 44(4): 781-789.
- 3 Elfwing R, Paulsson U, Lundberg L. Performance of SOAP in web services web environment compared to CORBA. Proc. of the Ninth Asia Pacific Software Engineering Conference (AP. SEC.02), 2002.
- 4 Lin KJ, Panahi M, Zhang Y, Zhang J, Chang SH. Building accountability middleware to support dependable SOA. IEEE Internet Computing, 2009, 13(3): 16-25.
- 5 吴斌. 企业服务总线中动态服务路由技术的研究[博士学位论文]. 济南: 山东大学, 2010.
- 6 <http://www.ibm.com/developerworks/cn/webservices/wsesbrouting>.
- 7 Apache, Servicemix, [http:// Servicemix.apache.org](http://Servicemix.apache.org).
- 8 Codehaus, Mule ESB. <http://mulesoft.org>.
- 9 Yin J, Chen H, Deng S, Wu Z, Pu C. A dependable esb framework for service integration. IEEE Internet Computing, 2009, 13(3): 26-34.
- 10 Xie JH, Bai XY, Chen B, Xiao SN. Review on enterprise service bus research. Computer Science, 2007, 34: 13-18.
- 11 Sun Y, Lian DB. An reliable and efficient dynamic routing model based on ESB. Computer System and Application, 2011, 20: 144-148.
- 12 Wu B, Liu SJ, Wu L. Dynamic reliable service routing in enterprise service bus. Proc. of the 3rd IEEE Asia-Pacific Services Computing Conference. IEEE Press. 2008.