

基于 SEDA 的企业服务总线的设计与实现^①

董 率^{1,2} 廉东本² 刘 鹏³ (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 信息化工程技术部 辽宁 沈阳 110171; 3.沈阳市环境信息中心 辽宁 沈阳 110171)

摘 要: 企业服务总线(ESB)是实现 SOA(Service Oriented Architecture)的基础设施, 现有的 ESB 在服务请求高并发的情况下性能严重下降。针对上述问题, 分析了 SEDA(Staged Event-Driven Architecture)模型的原理和 ESB 的功能需求, 将 SEDA 模型应用到 ESB 设计中, 设计并实现了一种基于 SEDA 模型的企业服务总线。对该 ESB 的性能进行测试和分析, 结果表明, 该企业服务总线在高并发情况下的性能有显著提高。

关键词: 企业服务总线; 面向服务构架; 阶段事件驱动模型

Design and Implementation of SEDA-Based Enterprise Service Bus

DONG Shuai^{1,2}, LIAN Dong-Ben², LIU Peng³

(1.Graduate University, Chinese Academy of Sciences, Beijing 100039, China; 2.Department of Information and Engineer, Shenyang Institute of Computing Technology, CAS, Shenyang 110171, China; 3 Shenyang Environmental Information Center, Shenyang 110171, China)

Abstract: The Enterprise Service Bus (ESB) is the infrastructure for the implementing of SOA (Service Oriented Architecture). The existing enterprise service bus has a serious decline in the performance in a high-concurrent situation. To address this issue, this paper analyses the principles of SEDA (Staged Event-Driven Architecture) model and the ESB functional requirements. The SEDA model is applied to the design of ESB, and a SEDA-based Enterprise Service bus is designed and implemented. According to the test and the analysis of the SEDA-based ESB, the SEDA-based ESB performs better in the case of high-concurrency.

Keyword: ESB; SOA; SEDA

1 引言

面向服务架构(Service Oriented Architecture, SOA)是一种业务驱动的 IT 架构方式, 支持对业务进行整合, 使其成为一种相互联系、可重用的业务任务或者服务^[1]。Gartner 在 1996 年第一次阐述了 SOA 的概念^[2]。目前, SOA 已经逐渐成为企业软件开发领域的主流技术。企业服务总线(Enterprise Service Bus, ESB)是用来整合应用和服务的一个灵活的基础

架构, 是 SOA 的一种实现方式。IBM 认为 ESB 是 SOA 的基础和核心^[3]。ESB 位于 SOA 的中心, 通过减少接口的数量、大小和复杂度使得 SOA 更强大。ESB 支持异构环境中的服务、消息, 以及基于事件的交互, 并且具有适当的服务级别和可管理性^[4]。

阶段事件驱动模型(Staged Event-Driven Architecture, SEDA)是加州大学伯克利分校研究的一套优秀的高性能互联网服务器架构模型^[5]。其设计

^① 收稿时间:2009-12-19;收到修改稿时间:2010-01-21

目标是：支持大规模并发处理、简化系统开发、支持处理监测、支持系统资源管理。

目前市场上已经有很多 ESB 产品，其中 IBM 和 Oracle 公司的 ESB 产品在市场上占有领先的技术地位，这些产品着重于集成技术，提供大量标准接口，强调集成能力，在管理上缺乏灵活性高效性，难以快速实施 ESB 应用。ServiceMix 和 Mule 是目前影响力比较大、使用比较广泛的两个开源 ESB 产品。ServiceMix 是一个基于 JBI 规范的开源服务总线。Mule 是一个轻量级的开源 ESB。它采用消息代理机制，在事件传递的过程中，Mule 消息传递也是通过 SEDA 实现的，但是 Mule ESB 的整体构架设计是紧密耦合的，而不是基于 SEDA 模型思想。ServiceMix 和 Mule 企业服务总线在服务请求高并发情况下的性能都会出现严重下降。

本文针对这一问题，将 SEDA 模型应用到 ESB 的设计和实现，提出一种基于 SEDA 的轻量级的企业服务总线的设计方案，能够显著提高企业服务总线在高并发情况下的性能。

2 企业服务总线的设计与实现

2.1 SEDA 模型介绍

SEDA 模型是针对处理大规模并发服务而提出的。它的核心思想是把一个请求处理过程分成几个阶段 (Stage)，对于消耗不同资源的每个阶段，其使用不同数量的线程来处理，每个阶段可以独立进行开发，阶段之间使用事件驱动的异步通信模式，通过事件队列 (Event Queue) 来进行通信。

每个 Stage 都配置一个线程池来处理本阶段的事件队列，对其线程池大小的动态调节，这种设计打破了多线程模型采用单一线程池的限制，采用多线程池和 Stage 配套使用，实现了对系统资源的细粒度控制，消除了基于线程的并发模型所带来的高负载，也有利于模块化和代码重用。SEDA 模型的一个阶段的结构有以下几个部分组成：

- ① 一个接受输入的事件队列；
- ② 一个应用开发者编写的事件处理器；
- ③ 一个控制器用于对执行过程进行控制。包括并发线程数量，批处理数量；
- ④ 一个线程池用于并发处理；每个阶段的输入通过事件队列获得。

每个阶段的输出会以事件形式推送到其他阶段的

事件队列中。每个阶段之间的这种连接关系由应用开发人员指定。图 1 描述了阶段的结构。

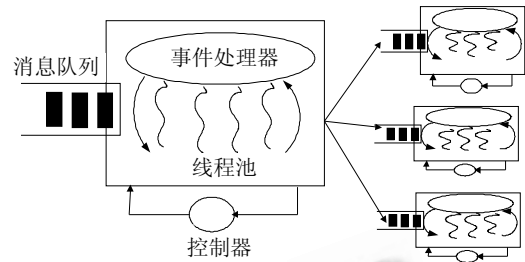


图 1 SEDA 阶段

2.2 企业服务总线功能需求

ESB 作为 SOA 的基础，提供了最基本的功能来保障 SOA 系统的运行，这些功能应该包含下列内容^[6]：

- 1) 提供位置透明性的路由和寻址服务；
- 2) 控制服务寻址和命名的管理功能；
- 3) 至少一种形式的消息传递范型（例如，请求/响应、发布/订阅等等）；
- 4) 支持至少一种可以广泛使用的传输协议。
- 5) 支持服务提供的多种集成方式，比如 Web 服务、异步通信、适配器等。
- 6) 一个开放且与实现无关的服务消息传递与接口模型，它应该将应用程序代码从路由服务和传输协议中分离出来，并允许替代服务的实现。

根据企业服务总线的功能分析，我们将企业服务总线的划分为以下功能模块：

- 1) 消息监听器：消息监听器负责监听客户端发来的服务请求消息，根据请求消息传输协议的不同将消息转发到不同的消息转换器。
- 2) 消息转换器：主要负责将不同协议的消息转换成标准消息，并将标准消息以消息上下文的形式传递到消息处理。
- 3) 消息处理器：通过调用消息处理引擎，根据服务代理的配置规则完成对消息上下文的处理。
- 4) 控制器：控制器的主要任务是对请求任务进行优先级管理、工作线程池的维护、协议转换器的注册与管理、负责消息处理器、消息路由器和注册管理器的控制与沟通。
- 5) 消息路由器：路由器负责确定消息的最终目的地址。
- 6) 服务注册管理器：外部服务注册管理器负责将

外部信息系统以服务的形式注册挂载到企业服务总线当中，并进行管理。

2.3 企业服务总线设计

根据对 ESB 的功能分析和 SEDA 原理，我们设计的基于 SEDA 模型的 ESB 的架构如图 2 所示。整个 ESB 系统共由六个模块组成：消息监听器、协议转换器、消息处理器、消息路由器、控制器和外部服务注册管理器。其中消息监听器、各个不同协议的消息转换器、消息处理器和消息路由器分别被设计为 SEDA 模型的不同阶段。

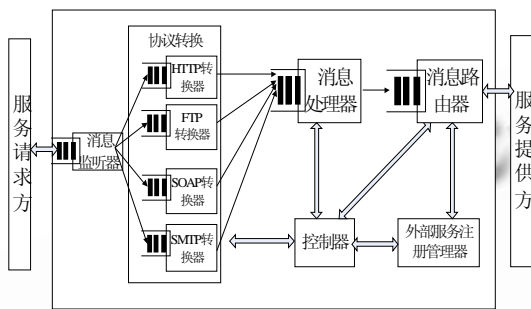


图 2 基于 SEDA 的 ESB 架构

服务总线的消息被封装成不同的 Event。消息监听阶段负责接收来自客户端的请求消息，并将消息封装到 Event 队列当中，监听阶段的事件处理器会针对不同的协议，将不同协议的 Event 分发到不同的协议转换器。协议转换器负责将不同协议的 Event 转换成标准格式消息的 Event，不同协议的消息转换由各自对应的阶段的事件处理器来完成，现在我们的 ESB 系统支持的协议有：HTTP、FTP、SOAP、SMTP 和 JMS 协议。协议转换模块是可扩展，我们可以针对特定的协议来开发特定的消息转换器，并将其封装成 SEDA 的一个阶段，将其配置到 ESB 系统中。消息处理阶段通过调用事件处理器，根据消息处理配置规则完成对 Event 的处理。

经过一系列的处理后，Event 被传递到路由阶段。消息路由阶段负责确定消息的最终目的地址（如图 3 所示），服务的最终目的地址可以是直接注册在配置文件中，可以是以 WSDL 文件的形式存储的。同时，路由还应该考虑到其所代理的服务的负载承受能力以及地址失效时框架本身的容错能力。最后路由器通过绑定真实服务的地址信息将服务请求发送到服务提供方。外部服务注册管理器负责将外部信息系统以服务

的形式注册挂载到企业服务总线当中，并进行管理。

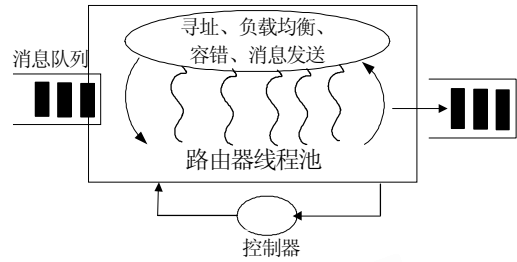


图 3 消息路由阶段

控制器作是服务总线的核心控制模块。在我们的设计当中，我们并不是针对每个 Stage 设计不同的控制器，而是将设计一个全局的集中控制器，控制器负责控制着每个阶段，主要包括对各个 Stage 线程池、事件的批处理进行维护和控制，控制器同时要维护不同事件的优先级，通常对于实时性要求高的同步请求事件会分配较高优先级，而对于实时性要求不高的异步请求事件则分配较低的优先级。

2.4 企业服务总线实现细节

具体实现当中，对于消息处理阶段，消息处理配置规则以配置文件的形式指明需要对 Event 进行的操作处理，配置文件设置处理序列 InSequence 和 OutSequence 分别对请求消息和响应消息进行处理，处理序列由多个处理节点 Mediator 构成，用户可以扩展 Mediator 接口实现自己的任务处理。一些常用的 Mediator 接口实现，比如对消息格式进行验证、进行消息格式的转换、对 SOAP 消息重新定向等已经在系统中实现，用户只需要通过配置文件即可完成。图 4 描述了消息处理节点的实现类图。

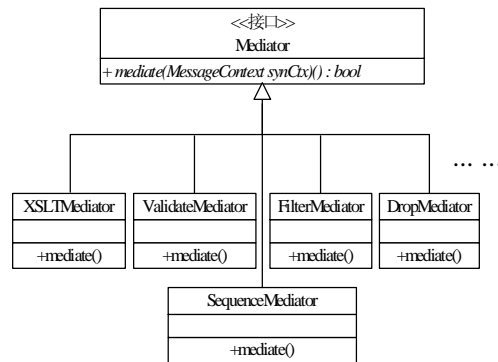


图 4 处理节点实现 UML 类图

实现中用 StageIF 接口作为一个 Stage 的抽象，

每个 Stage 的实现类都实现 StageIF 接口，如 ListenerStage、SOAPTransferStage、ProcessorStage、RouterStage。每个 Stage 拥有一个或多个事件队列，事件队列提供对 Event 的入队和出队操作，用 EventIF 和 EventQueueIF 这两个接口作为事件和队列的抽象。每个 Stage 都有一个事件处理器，事件处理器实现 Stage 的事件逻辑控制功能，用 EventHandlerIF 接口作为事件处理器的抽象，每个事件处理器通过实现 EventHandlerIF 来对 Event 进行特定业务逻辑处理，对应各个 Stage 的 EventHandlerIF 的实现类分别是 ListenerHandler、SOAPTransferHandler、ProcessorHandler、RouterHandler。系统 UML 类图如图 5 所示。

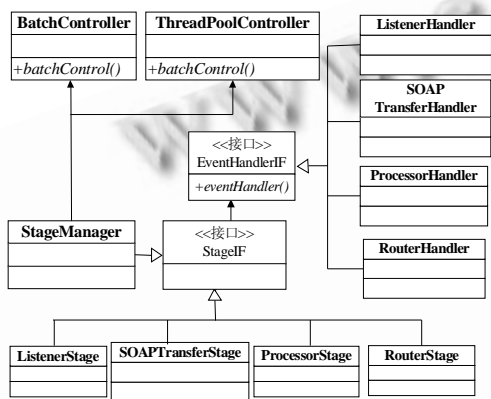


图 5 系统 UML 类图

控制器是服务总线的核心控制模块。控制器对 Stage 管理通过 StageManager 类来实现，StageManager 类包含对 ThreadPoolController 和 BatchController 的引用。ThreadPoolController 实现对 Stage 线程池的控制维护，BatchController 实现对每个事件处理器一次批处理事件数量的控制，当事件处理器比较繁忙时，减少一次批处理的事件数量，比较空闲时，增加批处理的事件数量。

在消息处理阶段，通常会出现有的消息事件需要及时处理，有的消息事件可以暂缓处理的情况。控制器将采用基于优先级的机制，对消息处理阶段的消息事件进行调度。控制器选择当前优先级最高的消息事件，交给处理线程进一步对消息事件进行处理。

3 性能测试与分析

使用了以下测试环境来测试企业服务总线的性

能：ESB 服务器：CPU: T3400 2.16G 操作系统：Windows XP 内存：1G JDK: 1.6 请求端客户机：CPU: Celeron 3.0G 操作系统：Windows XP 内存：1G JDK: 1.6 服务提供者：CPU: Celeron 3.0G 操作系统：Windows XP 内存：1G 数据库系统：SQL Server 2000 应用服务器：Tomcat、Jboss JDK: 1.6 测试环境如图 6 所示：

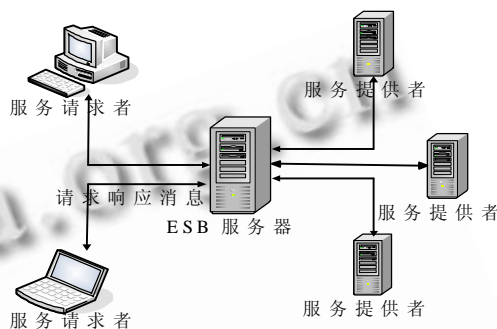


图 6 测试环境

使用以上测试环境，我们主要对基于 SEDA 的 ESB 的并发性能进行了测试。测试主要衡量的性能指标是 ESB 在不同数量的客户端请求消息的情况下，ESB 的平均响应时间。整个处理过程包括请求和响应消息的监听、处理和路由等阶段。测试方法为在客户端单位时间内发送不同数量请求消息，分别记录基于 SEDA 的 ESB 和非基于 SEDA 的 ESB 的响应时间。测试结果如图 7 所示：

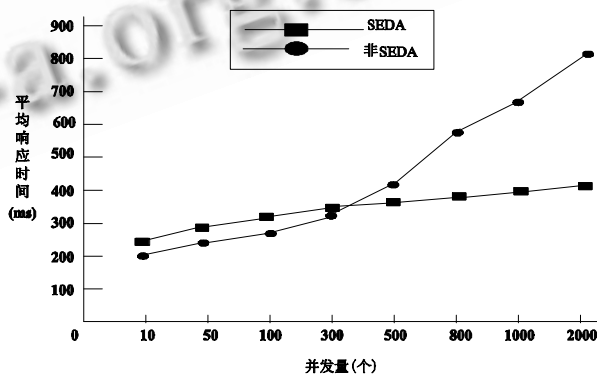


图 7 并发性能测试

从测试结果中可以看到，在轻量负载情况下，基于 SEDA 的 ESB 的处理性能优势并不是很明显，因为 SEDA 处理模式下系统需要维护额外的消息队列。但是随着并发负载的增多，SEDA 的性能明显优于没有使用 SEDA 方式的直接处理方式，这说明 SEDA 架构

在负载较重的情况下对消息处理过程的细粒度控制对ESB的性能改善是有效地,显著的提高了系统的并发处理能力,增加了ESB的吞吐量。

4 结语

本文运用SEDA模型的思想,对企业服务总线的基本功能进行深入分析的基础上,设计了一个基于SEDA的企业服务总线。通过对该企业服务总线进行测试分析可以证明,基于SEDA的企业服务总线能够完成一个企业服务总线基本功能,并且在服务请求高并发的情况下,其性能得到显著提高。今后的工作将根据实际需求,在此基础上进一步增强完善该企业服务总线的功能。

参考文献

1 Carter S.袁月杨 麻丽莉译. SAO & Web 2.0—新商业语言.北京:清华大学出版社,2007.

- 2 Gartner.Service-oriented Architecture Scenario. [2009-11-16].<http://www.gartner.com/resources/114300/114358/114358.pdf>.
- 3 IBM. Patterns: Implementing an SOA using an Enterprise ServiceBus. [2009-07-13]. <http://www.redbooks.ibm.com/redbooks/pdfs/sg246346.Pdf>.
- 4 邵欢庆,康建初.企业服务总线的应用.计算机工程,2007,33(2):220-222.
- 5 Welsh M. An architecture for highly concurrent, well-conditioned internet Service [Ph.D.Thesis]. Berkeley: Department of Electric Engineering and Computer Science, University of California at Berkeley, 2002.
- 6 Robinson R. Understand Enterprise Service Bus scenarios and solutions in Service-Oriented Architecture, Part1.[2009-08-21].<http://www.ibm.com/developerworks/webservices/library/ws-esbscen/>.