

## 遗留系统并行架构的设计与实现

罗丹,周波

(浙江大学 计算机科学与技术学院, 杭州 310027)

(luodan1987@zju.edu.cn)

**摘要:**面向服务的体系架构(SOA)为遗留系统的再工程提供了解决方案,使得遗留系统可以支持分布式应用环境,但是由于技术的陈旧和架构的局限性,无法支持多线程、并行处理以及内存泄露等问题依旧在部分遗留系统中存在,极大地限制了它们的应用。为了解决这几个问题,通过深入分析研究 Windows 通信基础(WCF)的通信机制,提出了一种并行架构,对 WCF 的基本架构进行了改造,即在默认的体系架构中添加一层服务控制器,用来在客户端和服务端之间传递消息和选择服务,很好地解决了这几个问题,并在某大型金融软件中得到了应用。

**关键词:**遗留系统;并行;Windows 通信基础;内存泄露

**中图分类号:** TP311 **文献标志码:** A

## Design and implementation of parallel architecture for legacy system

LUO Dan, ZHOU Bo

(College of Computer Science and Technology, Zhejiang University, Hangzhou Zhejiang 310027, China)

**Abstract:** Service-Oriented Architecture (SOA) provides a solution for reengineering legacy systems to support distributed application environment. But due to the limitation of technology and architecture involved, issues such as non-support of a multi-threaded environment, memory leaking, non-support of parallel environment still exist in some parts of the legacy systems, which restrict their applications immensely. Through analyzing and doing research on the communication mechanism of Windows Communication Foundation (WCF), a parallel architecture was proposed to solve these problems. The architecture improved the default architecture of WCF by adding a service controller to it, which could deliver message between the client and the server and selected service for clients. The proposed architecture has already solved these problems and been applied in a large financial system.

**Key words:** legacy system; parallel; Windows Communication Foundation (WCF); memory leaking

### 0 引言

随着商业环境的快速变化,以及技术的不断创新,遗留系统面临着越来越大的生存压力,主要表现为系统结构混乱致使维护费用居高不下、设计框架陈旧无法跟上分布式环境的需求以及开发技术落后带来潜在的内存泄露等。近年来,随着遗留系统再工程研究的不断深入,部分危机已得到了缓解。

目前,基于面向服务的体系结构(Service-Oriented Architecture, SOA)架构的再工程方法主要是将遗留系统封装成 Web Service 对外提供服务,然后引入 Wrapper 层提供 Web Service 的具体实现,Wrapper 层和遗留系统的代码进行交互以实现重用,所有与外界的通信都经由代理接口转换,应用这种架构可以让遗留系统支持分布式环境的需求,但是潜在的内存泄露依旧存在,这是因为这种方法直接重用了遗留系统的代码。

本文提出了一种基于 Windows 通信基础(Windows Communication Foundation, WCF)的并行架构,重点研究遗留系统封装成 Web Service 之后如何在单机上并行处理分布式请求、如何支持多线程处理以及如何解决内存泄露等问题,并使得这个架构具有通用性,即不依赖于具体的应用系统。

### 1 系统架构

图1的架构实现了本文提及的几个目标:支持服务的并行处理、支持多线程、解决服务的内存泄露。相比一般的 WCF 应用,该架构在客户端和服务端通信的中间引入了一个服务控制器,让客户端和服务端间接地进行通信,如图2所示。

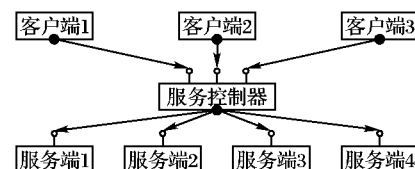


图1 核心体系架构

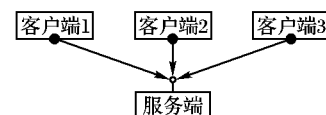


图2 WCF 体系架构

服务控制器维护一个服务实例池,在单机上根据通信状况动态开启若干个服务,协调客户与服务的通信,通过将客户的请求分配到不同的服务实例上执行以实现客户需求的并行处理,提高响应速度,同时实现服务对多线程的支持;另外,服

收稿日期:2010-07-30。

**作者简介:**罗丹(1987-),男,江西新余人,硕士研究生,主要研究方向:NET应用、数据库;周波(1969-),男,浙江文成人,副教授,主要研究方向:超大规模信息系统、数据库管理系统、CIMS、数据仓库、数据挖掘、金融信息工程。

务控制器时刻监测池中服务的运行状态,有选择地关闭服务,以解决服务内存泄露的问题。

## 2 关键技术

### 2.1 SOA

SOA 是由 IBM、Microsoft 等公司倡导的一种新型的企业级系统架构,是一种大型的分布式系统模型<sup>[1]</sup>。它将应用程序的不同功能单元抽象为一系列粗粒度的业务服务和业务流程,并将服务通过定义良好的接口和契约联系起来。接口和契约采用中立、基于标准的方式进行定义,它独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在不同系统中的服务可以以一种统一并通用的方式进行交互、相互理解<sup>[2]</sup>,使得服务的请求者和提供者之间高度解耦。

SOA 通过服务提供者、服务请求者和注册中心等实体之间的交互实现服务的共享和调用<sup>[2]</sup>,以服务或者组件形式出现的业务逻辑可以被共享、重用和配置,能很好地满足遗留系统的要求,加快再工程的开发进度。

### 2.2 WCF

WCF 是 Microsoft 为构建面向服务的应用提供的分布式通信编程框架,实际上可以看做是微软对 SOA 架构的一种实现。它提供了对可靠性、事务性、并发管理、安全性及实例激活等技术的有力支持,它们均依赖于基于拦截机制的 WCF 体系架构<sup>[3]</sup>。如图 3 所示,客户端经过路径 A 到达服务端是 WCF 的默认架构,而路径 B 服务于文中所提的架构,是对 WCF 默认架构的扩展。

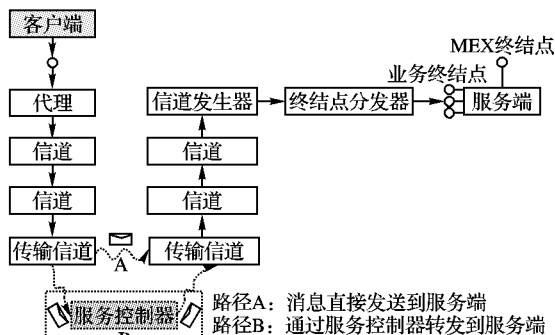


图3 WCF 体系架构

WCF 基于消息进行通信,从这个角度讲,WCF 可以看成是对消息进行发送、传递、接收的基础工具<sup>[4]</sup>。在客户端,代理把调用栈帧(Stack Frame)序列化到消息中,并将消息通过通道链向下传递,不同的通道执行一个特定的任务(比如对消息的编码、加密、签名等),最后根据配置的传输方式发送消息给服务端;在服务端,消息同样通过通道链进行传输,它会对消息执行服务端的调用前处理(比如对消息进行解密、解码等)<sup>[4]</sup>。通道链的组成与结构主要依赖于绑定。

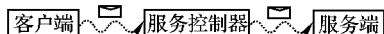


图4 消息传输路线 B



图5 消息传输路线 A

如图 4 所示,在 WCF 体系架构的基础上,增加了一个服务控制器,客户端的请求消息不再直接发送到服务端,而是通过服务控制器进行转发,这个服务控制器正是文中所提架构的核心,如图 5 所示。服务控制器收集客户端的请求消息,经过处理转发给服务端,然后服务控制器再将服务端的响应消

息返回给客户端。

## 3 服务控制器的设计与实现

服务控制器实质上是一个多线程的 WCF 服务,它由实例提供者、消息转发器、服务实例池、服务清除器以及服务代理 5 部分构成。它向外只暴露消息转发器的接口服务,消息转发服务是服务控制器的核心。图 6 为服务控制器处理客户端请求的时序图及其各组成部分的职责。

以下为处理一个客户端请求的具体步骤。

1) 服务控制器启动后,立即对服务实例池进行初始化,根据配置文件预先启动若干个服务进程,并将各个服务代理存储在实例池中,同时启动请求匹配线程<sup>[5]</sup>和服务进程清理线程。

2) 当请求到达时,WCF 分发器<sup>[6]</sup>调用实例提供者的 GetInstance 函数以获取消息转发器的实例,实例提供者首先将请求添加到服务实例池中,唤醒匹配线程,然后创建一个消息转发器实例返回给 WCF 分发器。

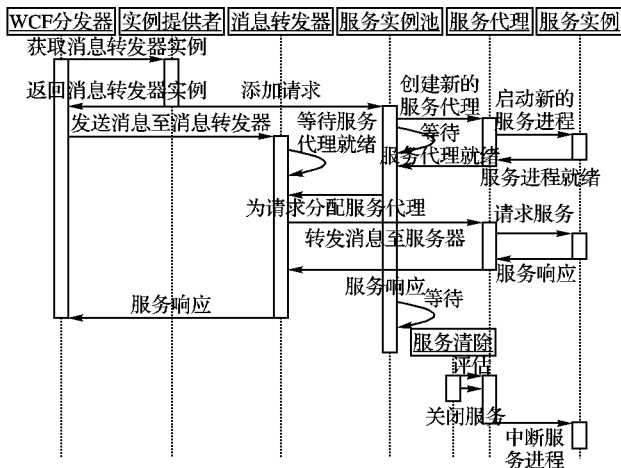


图6 客户端请求处理时序图

3) 消息转发器在接收到请求消息后,首先判断自己的服务代理是否就绪。若就绪,直接将请求消息通过服务代理发送给服务端;否则,等待匹配线程为该请求分配服务代理。

4) 匹配线程被唤醒后,会遍历服务实例池中所有未匹配的请求和可用的服务代理,一旦某个请求匹配服务成功,该请求相对应的消息转发器就会被唤醒,继续完成其转发工作。在匹配过程中,如果没有找到可用的服务代理,匹配线程会等待新服务代理的创建。新服务代理的创建由另一个线程完成,一旦创建完毕,匹配线程会被唤醒。因为支持多个服务实例,所以某客户端的多个请求同时到达时,可以保证这些请求得到并行处理,提高服务的响应速度。

5) 服务端的响应消息最后再通过消息转发器返回给客户端。

6) 服务清理器以轮询的方式对服务实例池中的服务实例进行评估,一旦满足清除的条件,相应的服务会被强制清除,以解决服务内存泄露的问题。

### 3.1 实例提供者

实例提供者用来为 WCF 分发器提供服务实例,此处提供消息转发器的实例。WCF 默认支持 3 种实例提供方式<sup>[3]</sup>:单调服务、会话服务和单例服务。由于在提供消息转发器实例时需要额外的操作,本文对 WCF 的实例提供机制进行了扩展<sup>[7]</sup>,实现了自定制的实例提供者,不过本质上实例提供者

遵循的是单调服务的规则。它实现了 `IInstanceProvider` 接口, 在函数 `GetInstance` 中, 除了返回消息转发器的实例外, 最重要的是唤醒匹配线程, 为新添加的请求匹配服务代理。

### 3.2 消息转发器

消息转发器主要负责在客户端与服务端之间传递消息。消息转发器在截获到客户端的消息后, 只要相应的服务代理就绪, 该消息就会通过就绪的服务代理发送到服务端上, 最后消息转发器将服务端的响应消息转回给客户端完成一次请求—响应的过程。

有两种方式可以让消息转发器截获到客户端发往服务端的请求消息<sup>[8]</sup>: 一种是客户端在请求服务时, 主动将请求消息发送至服务控制器(本文采用此方案); 另一种是将服务控制器的监听地址修改为服务的地址, 同时对服务的监听地址作相应的修改。

消息转发器提供一个 `Forward` 操作, 这也是服务控制器暴露的唯一的, 它的参数和返回值都是 `Message` 类型, 这样, 可以保证 `Forward` 能处理任何请求和返回, 因为 WCF 中服务的调用最终是基于 `Message` 的<sup>[9]</sup>。

另外, WCF 的分发器在收到消息之后, 需要在服务控制器中选择一项操作进行调用<sup>[6,9]</sup>, 默认情况下分发器是根据传入消息的 `Action Header` 进行选择的, 一般按如下规则进行匹配: `Contract Namespace/Contract Name/Action = action in SOAP header`。如果将操作的 `Action` 设为“\*”, 则意味着无论 SOAP Header 中 `action` 是什么, 都将交给该操作来处理。本文中, 消息转发器需要接收客户端任何类型的请求, 而 `Action` 属性正好可以满足要求。

### 3.3 服务实例池

服务控制器中, 除了消息的传递由消息转发器完成外, 绝大部分工作由服务实例池完成, 包括启动匹配线程、完成匹配工作、唤醒消息转发器、为服务提供缓存机制<sup>[10]</sup>等。而其中最重要的工作是为服务提供缓存机制, 它维护一个实例对象池, 以提高系统的性能, 减少系统的响应时间。

在匹配工作完成之后, 实例池依据以下任意一个条件决定是否启动新的服务代理: 服务代理的总数少于指定值(配置文件中设置); 匹配的时候没有找到可用的服务代理, 并且服务代理总数没有超过指定值。

### 3.4 服务代理

服务代理主要负责启动服务进程、关闭服务进程以及创建服务通道<sup>[4]</sup>。因为服务进程是动态启动的, 因而服务地址不确定, 也就是说, 事先无法明确地告诉客户端服务的地址和支持的协议是什么, 所以在 3.2 节采用第一种方案来截获客户端的请求消息。

新的服务进程都是通过服务代理启动的, 这限制了新启动的服务只能驻留在本机, 故服务控制器和服务之间使用命名管道的传输方式进行通信。

### 3.5 服务清除器

服务清除器主要用来解决内存泄露问题, 它以轮询方式作为一个后台线程执行, 如果服务实例满足如下任意一个条件, 服务实例将被强制关闭。服务实例可用, 但是其内存消耗超过指定值(配置文件中设置); 服务实例可用, 但是启动的服务实例数超过指定值; 服务实例可用, 但是其运行总时间超过指定值。

## 4 实例研究

该架构在某大型金融项目中得到了应用, 它对一个投资

会计管理遗留系统进行再工程。遗留系统是个 Client/Server 结构的单机系统, 问世已有 20 来年, 核心代码基于 MFC 开发, 再工程项目的目标是将其完全移植到 SOA 系统框架中, 项目已经应用 WCF 技术将遗留系统各个功能模块封装成 Web Service, 为客户提供分布式的服务。但是, 在不断地对这个庞大的遗留系统进行封装再工程时, 发现服务自身存在内存泄露的问题, 且无法从代码层面上消除内存泄露, 导致服务要不断地被重启才可以应付客户频繁的请求, 影响了服务的稳定性; 另外, 该遗留系统不支持多线程环境, 因而简单地暴露为 WCF 服务, 尚无法支持客户请求的并行处理, 极大地影响了服务的响应时间。

引入本文所提的并行架构后, 通过在单机上开启多个服务来响应客户的请求, 克服了并行的困难, 提高了服务的响应速度; 通过不断地监测单机上各个服务的运行状态, 很好地解决了内存泄露的问题。

## 5 结语

本文针对一大型金融软件在应用 WCF 进行再工程中暴露出来的各种问题, 如遗留系统自身潜在的内存泄露、无法支持并行处理、无法支持多线程等问题, 提出了一套通用的并行架构, 即通过在默认的 WCF 体系架构中添加一层服务控制器, 用来在客户端和服务端之间传递消息和选择服务, 几个问题都得到了较好地解决。

但是这套架构提供的服务控制器只能控制单机上的服务, 无法充分利用 WCF 的分布式架构, 下一步的工作需要研究如何将客户端的请求转发到网络上不同的服务节点上(而不再仅仅限于单机上转发), 实现路由的功能, 并进行拥塞控制和路由选择等。

### 参考文献:

- [1] 许鹏. SOA 架构下 Web Services 实现的企业遗留系统重用研究[D]. 合肥: 合肥工业大学, 2008.
- [2] 毛新生. SOA: 原理·方法·实践[M]. 北京: 电子工业出版社, 2007: 4-19.
- [3] McMURTY C, MERCURI M, WATLING N, *et al.* Windows communication foundation unleashed[M]. Indianapolis: Sams Publishing, 2007.
- [4] LOWY J. Programming WCF services[M]. Sebastopol: O'Reilly, 2007.
- [5] Microsoft Corporation. Multithreading and asynchronous programming in Web applications[EB/OL]. [2010-06-01]. <http://msdn.microsoft.com/library/en-us/dnmpag/html/DiforWC-CH06.asp>.
- [6] 蒋金楠. WCF 后续之旅(3): WCF service mode layer 的中枢——dispatcher[EB/OL]. [2010-06-01]. <http://www.cnblogs.com/artech/archive/2008/07/15/1243092.html>.
- [7] 蒋金楠. WCF 后续之旅(4): WCF Extension Point 概览[EB/OL]. [2010-06-01]. <http://www.cnblogs.com/artech/archive/2008/07/16/1243956.html>.
- [8] 蒋金楠. WCF 后续之旅(13): 创建一个简单的 WCF SOAP Message 拦截、转发工具[上篇][EB/OL]. [2010-06-01]. <http://www.cnblogs.com/artech/archive/2008/09/01/1280939.html>.
- [9] 蒋金楠. WCF 技术剖析(卷1)[M]. 北京: 电子工业出版社, 2007.
- [10] 蒋金楠. WCF 后续之旅(10): 通过 WCF Extension 实现以对象池的方式创建 Service Instance[EB/OL]. [2010-06-01]. <http://www.cnblogs.com/artech/archive/2008/08/05/1260594.html>.