

基于客户端代理的透明服务接管技术研究

何晓忠¹, 黄永忠¹, 杨 岳¹, 南 煜²

(1. 解放军信息工程大学信息工程学院计算机科学与技术系, 郑州 450002;

2. 解放军信息工程大学理学院电子工程系计算中心, 郑州 450001)

摘 要: 在服务消费过程中, 为保证可用性和服务质量, 需对服务调用请求实施失效监控和透明失效接管, 而基于服务端集中式代理的监控与接管方式容易引起服务端过载, 致使服务系统的整体可用性和服务质量下降。针对该问题, 提出一种基于客户端代理的透明服务接管方案, 以分布式、基于服务描述自动生成的客户端代理取代集中式的服务端代理。该方案在解决集中式代理过载问题的同时能够保证较高的开发效率。

关键词: 代理; 透明; 服务接管; 客户端

Research on Client-side Proxy-based Transparent Service Take-over Technology

HE Xiao-zhong¹, HUANG Yong-zhong¹, YANG Yue¹, NAN Yu²

(1. Dept. of Computer Science and Technology, Institute of Information Engineering, PLA Information Engineering University, Zhengzhou 450002, China; 2. Computing Center, Dept. of Electronic Engineering, Institute of Science, PLA Information Engineering University, Zhengzhou 450001, China)

【Abstract】 In service oriented architecture, transparent service take-over is the main method of reducing the probability of dissatisfaction of service requesting from the consumer when the service is unavailable. For the purpose of transparent service take-over, the client-side and proxy-based approach can avoid the situation of the whole system service availability decreasing, which is probably caused by overloading when using the centralized one on the server side. In the client-side and proxy-based approach, automatic generation of proxy source code from service description can obviously improve the efficiency and decrease the difficulty of the developing process.

【Key words】 proxy; transparent; service take-over; client side

1 概述

Web 服务技术以其松耦合、晚绑定、自治、平台无关和基于开放标准等诸多优点^[1], 成为面向服务架构的主要实现技术^[2]。对于很多企业而言, 保证其服务系统中服务的可用性相当关键, 短时间服务失效, 都可能会造成巨大的损失。然而, 因网络、节点和服务自身等诸多因素的存在, 服务的失效无法完全避免, 需要一种方法来确保失效存在的系统的正常运行。

副本技术被广泛地应用于关键服务的容错^[3-4], 在服务失效时, 使用副本接管失效的服务, 以满足服务的消费需求。多数服务系统在服务端进行服务失效接管, 为每一次服务调用进行消息缓存并运行相应的监控和接管逻辑。在消耗大量服务端的资源的同时加重了服务端的负载, 在高并发请求下, 服务端过载的可能性大大增加, 此外, 缓存于服务端的服务调用消息既要保证可靠性和完整性, 又要保证快速的访问和存储, 代价较高。

本文研究一种使用基于服务描述自动生成的客户端代理实现透明服务接管的方案, 避免了上述的服务端过载和消息缓存问题。

2 基于代理/中介的透明服务接管

不管是端到端模型^[5]、SOA 三角模型^[5], 还是其他的模型, 服务消费环节均存在服务失效问题。为提高系统可用性,

保证在服务失效的情况下, 服务的消费请求被满足, 需要在服务消费过程中引入失效接管机制。

通过在消费方和供应方之间加上代理、中介或者中心集线器^[6](下文以 P 统称), 由 P 替代消费方, 直接与服务供应方通信, 在服务失效的情况下, P 为消费方寻找等价的替代服务, 接替失效的服务。

多数服务系统在引入 P 时, 使用集中式的 P, 如图 1 所示, 所有的服务消费均通过 P 委派到真正的服务供应方。使用这种方案时, 在面临大量并发的服务消费请求时, P 的负载能力面临较大考验, 一旦负载超过 P 能够承受的阈值, 或者 P 自身出现问题, P 将无法正常工作。

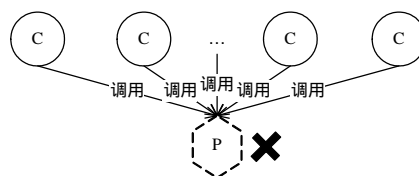


图 1 集中式 P 失效

作者简介: 何晓忠(1984 -), 男, 硕士研究生, 主研方向: 分布式系统; 黄永忠, 教授、博士、博士生导师; 杨 岳, 硕士; 南 煜, 讲师、硕士

收稿日期: 2010-02-04

E-mail: mailxiaozhong@126.com

为避免这一情况发生,可以使用多 P 备份。但即使使用多 P 备份,由于服务调用时需保存调用消息,以备失效时可以向接管的服务发送原始消息,在高并发下,服务端需保存大量消息,保证消息的快速访问和完备性是个大挑战。此外,P 的失效,容易造成状态丢失,这会导致备份 P' 与失效 P 的原先状态失去同步,进而破坏整个系统的一致性,如图 2 所示。要让备份 P' 恢复失效的 P 的状态需要引入状态同步机制,代价高昂,容易成为瓶颈。

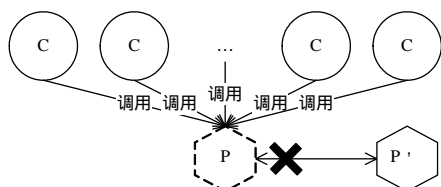


图 2 集中式 P 备份状态失同步

3 基于客户端代理的透明服务接管

基于客户端代理的透明服务接管,将代理由集中的方式变为分布方式,每个服务消费方均有代理,各代理互不影响,如图 3 所示。

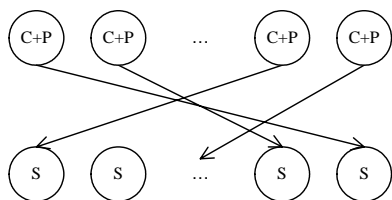


图 3 基于客户端代理模型

图 4 是基于客户端代理的透明服务接管方案的代理设计方案。

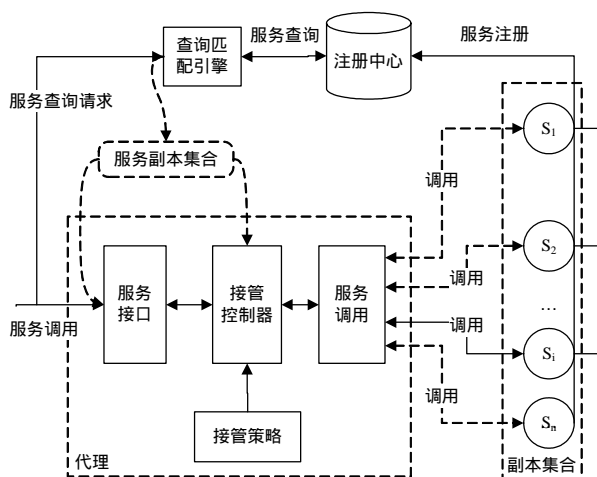


图 4 客户端代理的设计

在该方案中,服务的消费和失效接管过程如下:

- (1) 服务消费方通过服务查询匹配引擎向服务注册中心发起服务查询请求,得到所有满足条件的服务副本集合。
- (2) 服务消费方从获得的服务副本集合中选择一个服务副本作为目标服务,并通过代理的服务接口发起针对该服务消费请求。
- (3) 服务接口向接管控制器发起针对目的服务的调用请求。
- (4) 接管控制器在接到服务调用请求后向目的服务所在服务供应商发起服务调用并启动监视过程。

(5) 如果目的服务被成功调用并返回结果,接管控制器将结果通过服务接口返回给服务消费方,服务调用过程结束。

(6) 否则,本次服务调用失败,此时代理并不直接向消费方报告该次调用失败的情况,而是启动服务接管过程,依接管策略,从备选服务副本集中选择另外一个副本来接替失效的服务,作为新的目的服务,重启调用过程,直到服务消费方的服务请求得到满足时结束,或者在所有服务副本均失效的情况下,报告服务调用失败。上述服务失效接管的过程对消费方是透明的。

每一个服务消费方均有自己的代理,且每个消费方的服务需求动态变化的,若按照每个代理的需求编写代理的代码显然是效率低下的,需要有更有效的方法。

4 基于 Web 服务描述的代理自动生成

基于 Web 服务描述的自动代理生成的原理是对服务的描述文档进行解析,从解析获得的服务描述信息动态生成代码。生成代码时,将服务调用接口、服务调用实现和服务透明接管逻辑分开,以图 5 中 proxy 模式进行组织。

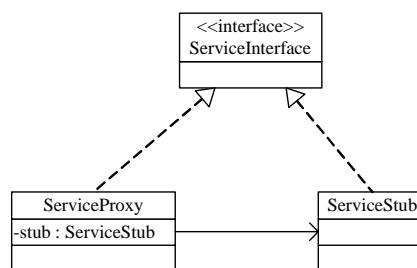


图 5 从服务描述生成代理的 UML 图

分为服务调用接口(ServiceInterface),实现该接口的服务调用类(ServiceStub)和服务代理类(ServiceProxy)3 个部分,服务的调用由代理类委派到调用类,代理类则对调用的过程封装透明服务失效接管过程。

基于服务描述自动生成代理的过程如下:

- (1) 从服务注册中心查询获得目的服务的 WSDL 描述文档。
- (2) 解析目的服务的描述文件。获取服务接口、参数和返回值信息,生成相应的服务接口 ServiceInterface。
- (3) 生成进行服务调用的 ServiceStub 类。该类实现了第(2)步生成的服务接口,为 ServiceStub 类增加初始化和设置目的服务的代码,以及服务调用方法逻辑。由初始化代码指定目的服务,该指定服务将被调用。
- (4) 生成服务代理 ServiceProxy 类。该类实现第(2)步生成的服务接口,通过其 stub 属性与 ServiceStub 关联,在实现 ServiceInterface 的方法时,并不直接调用目的服务,而是通过 stub 属性,调用 ServiceStub 类的方法实现的,这一调用过程封装了对 stub 方法的调用的代码以及监控和失效接管逻辑,在服务失效或者超时情况发生时,对 stub 的目的服务进行重新配置,并重启调用过程。
- (5) 编译生成。

5 接管策略

服务接管策略指的是服务失效后,用于失效接管的备选服务的选取策略,有随机策略和顺序策略。随机策略指的是每次出现服务失效时,从候选集中未失效的服务中任选一个作为接替的服务;顺序策略指的是对给定的副本序列,从头(下转第 276 页)