# SCF框架设计&使用说明文档

www.spat.org

2016-04-04

peida 撰写

## 前言

在互联网行业内,业务发展早起，基于Java Web 的经典MVC框架SSH或者SSI(Spring +Structs/SpringMVC +Hibeenate/iBatis）+Tomcat 的架构相对技术比较单一，学习成本比较低，开发上手快，测试部署和运维也比较简单。



但随着公司业务的不断发展，系统需要支撑用户数量和需求的也会逐步增多，同时团队的规模也会越来越大，单纯的基于Java Web的框架（例如：SSH，SSI等）MVC框架将面临很多棘手的问题，会严重的影响业务的发展。

传统的垂直应用架构面临的一下4个方面的棘手问题：

1. 代码重复率高：一些业务层的公共功能，被多个模块重复开发，导致研发成本上升，代码质量下降，架构开始腐化，为后续系统维护和增加新功能的带来巨大的挑战。
2. 需求变更困难：由于长流程无法有效拆分、代码重复率高等因素，导致每次需求变更都影响一大片，需要大量的回顾测试才能保证质量，需求的交付周期被不断拉长。
3. 部署效率低：业务没有拆分，一旦有一个业务变更，就需要重新编译打包；巨无霸应用包含功能模块过多，编译，打包时间较长，一旦编译出错，需要定位和修改错误后在编译，耗时较长。
4. 学习成本高：业务流程有一长串方法串联起来，臃肿冗长。随着业务的发展和需求变化，代码的不断迭代和变更，最后会形成一个个功能孤岛，原有开发负责人离职或者变动，新人接手周期变长，并且系统运维将变得非常困难。

针对以上问题，我们需要一个基于SOA的RPC框架来解决单块的传统垂直应用架构问题，来提高代码利用率，需求快速响应能力，降低系统部署难度，及团队学习成本等。

基础的SCF框架主要功能有以下四部分组成：

1. 客户端
2. 协议
3. 序列化
4. 服务端

**文档使用约定**

本文档主要针对SCF的使用进行讲解，主要Java客户端，Java服务器端的开发。本文中提到的SCF为SCF本身，包括客户端、服务端、协议、序列化，SCF容器指可以部署服务的容器本身。

**Java版依赖：**

org.spat.scf.protocol-3.0.0-SNAPSHOT.jar

org.spat.scf.serializer-3.0.0-SNAPSHOT.jar

org.spat.scf.server-3.0.0-SNAPSHOT.jar

org.spat.scf.client-3.0.0-SNAPSHOT.jar

# 第1章 SCF简介

## 1.1 什么是SCF

SCF是服务通讯框架(Service Communication Framework)支持跨平台具有高并发、高性

能、高可靠性，并提供异步、多协议、事件驱动的中间层服务框架。

## 1.2 SCF组成

SCF主要由4部分组成 ：

□ SCF Server：服务容器，用于宿主开发人员所开发的SCF服务，接收和处理来自客户端的请求。

□ SCF Client：多种平台客户端，调用者就像在调用本地接口一样方便，同时还提供了负载均衡，容错等机制。

□ SCF Serializer：提供了跨平台的二进制序列化解决方案。

□ SCF Protocol：分为传输协议和数据协议．传输协议用于进行数据传输，数据协议用于请求和响应结果的内容数据。

## 1.3 特点

□ **高性能**：客户端和服务端特定的通讯模型，序列化组件对序列化和反序列化性能以及结果字节数组大小的严格控制，通讯协议的针对性设计等使得SCF 比.net平台的WCF，Java平台的EJB、RMI，跨平台的WebService等性能都要好。在追求性能的同时支持客户端的HA（High Availability），容错机制为服务提供了良好的可靠性保证。

□ **跨平台：**由于特定系统的需要，会使用特定的语言或平台来开发，这么一来会存在多个平台的情况，比如有java、.net、c++等，想要实现这些系统的简单高效的跨平台调用是一个非常麻烦的事情，SCF通过为不同平台提供不同的客户端来实现跨平台，目前已有

java、.net、c++等平台的客户端。服务开发人员在开发服务时无需关心这些细节，所开发

出来的服务就能很好的支持跨平台调用。

□ **便捷开发：**这是SCF 设计的主要目标之一，开发人员不需要了解网络通讯，不需要关心如何跨平台，不需要知道如何塔建一个服务集群以及怎么做HA，这些框架都做了封装，需要做的只是定义一个接口，实现接口方法，标注注解。可以便捷的使一般程序员能够开发出优秀程序员一样高效、安全、稳定的服务。对于服务调用者也是非常简单的，所有服务开发人员不需要了解的东西调用者同样也不需要了解，对调用者来说调用远程服务的方法就像是调用本地方法一样。

□ **高扩展性：**当服务有压力时可以随时加服务器，增加服务节点来分担压力，客户端的负载均衡模块可以根据服务结点的压力情况来调整各结点的权重并平衡压力。

□ **安全性：**SCF可以对调用者进行权限授权，不同的调用者只能调用对他授权的方法，这对有些对外暴露的服务又想对访问进行授权将会非常有用。