**RPC框架设计**

1. **背景**

RPC(Remote Procedure Call) --远程过程调用，是分布式系统中广泛使用的一种过程调用方式。RPC使得开发包括网络[分布式](https://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F" \t "https://baike.baidu.com/item/%E8%BF%9C%E7%A8%8B%E8%BF%87%E7%A8%8B%E8%B0%83%E7%94%A8%E5%8D%8F%E8%AE%AE/_blank)多程序在内的应用程序更加容易。

RPC采用客户机/服务器模式。请求程序就是一个客户机，而服务提供程序就是一个服务器。首先，客户机调用进程发送一个有进程参数的调用信息到服务进程，然后等待应答信息。在服务器端，进程保持睡眠状态直到调用信息到达为止。当一个调用信息到达，服务器获得进程参数，计算结果，发送答复[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF/111163" \t "https://baike.baidu.com/item/%E8%BF%9C%E7%A8%8B%E8%BF%87%E7%A8%8B%E8%B0%83%E7%94%A8%E5%8D%8F%E8%AE%AE/_blank)，然后等待下一个调用信息，最后，客户端调用进程接收答复信息，获得进程结果，然后调用执行继续进行。

1. **现有解决方案**

RPC需要将函数标识、函数参数等进行编码，通过底层传送协议(TCP、UDP、HTTP/HTTPS等)传送到远端，远端进行解码，获取本地调用的函数标识、参数；本地执行后将执行结果编码，通过传送协议回送给调用者。

现有RPC实现方案有Sun 公司主导的开源 ONC RPC (后修订为IETF标准，参见RFC1833)以及Google公司主导的基于protobuf实现的gRPC等。

无论哪种解决方案，RPC的实现均需要考虑RPC消息及参数的编码(如：XDR、proto文件语法)、接口定义框架代码的生成工具等(如：rpcgen)。

1. **问题**

在基于RPC进行业务软件开发、实施时，需要考虑以下问题：

1. RPC服务代码和客户代码可能会使用不同的开发语言实现；
2. 在实际开发、部署过程中，函数设计/服务功能 也面临着新增或者修订；
3. RPC消息编码需要兼顾故障定位的便捷性和编解码的效率。

请结合业界开源RPC的实现方案，结合上述主要诉求；设计一套可能的RPC实现框架。

要求：请您以Word输出整体运作方案，并将其中要点以PPT形式进行输出，在极致挑战环节进行宣讲。