# Implementing Remote Procedure Calls

远程过程调用（RPC）似乎是用于以高级语言编写的程序之间通过网络提供通信的有用范例。 本文介绍了一种提供远程过程调用功能的程序包，此类程序包设计人员面临的选项以及我们做出的决策。 我们描述了RPC机制的整体结构，用于绑定RPC客户端的工具，传输级通信协议以及一些性能度量。 我们包含一些用于实现高性能并最大程度减少具有许多客户端的服务器计算机上的负载的描述。

这个想法有很多吸引人的方面。一种是**简洁明了的语义**：分布式计算的构建和正确性变得更加容易。另一个是**效率**：调用过程足够简单，使得通信速度相当快。第三点是**通用性**：在单机计算中，进程通信是算法各部分之间进行通信的重要部分，这样RPC可以应用到很多场景中。

RPC工具设计人员面临的主要问题包括：在存在机器和通信故障的情况下调用的精确语义；在（可能）没有共享地址空间的情况下，包含地址的参数的语义；将远程调用集成到现有的（或将来的）编程系统中；绑定（呼叫者如何确定被呼叫者的位置和身份）；在呼叫者和被呼叫者之间进行数据传输和控制的合适协议；以及如何在开放的通信网络中提供数据完整性和安全性（如果需要）。

## 设计目标

我们的RPC项目的**主要目的**是简化分布式计算。以前，在我们的研究社区中发现，交流程序的构建是一项艰巨的任务，仅由选定的交流专家组的成员来承担。即使是具有丰富系统经验的研究人员，也很难获得使用现有工具构建分布式系统所需的专业知识。这似乎是不可取的。我们拥有一个非常庞大，非常强大的通信网络，许多强大的计算机以及一个使构建程序相对容易的环境。现有的通信机制似乎是制约分布式计算进一步发展的主要因素。我们希望通过提供几乎与本地过程调用一样轻松的通信，将鼓励人们构建和试验分布式应用程序。我们希望，RPC将消除不必要的困难，而只剩下构建分布式系统的基本困难：计时，组件的独立故障以及独立执行环境的共存。

我们有两个次要目标，希望能够支持我们的目标。 我们希望使RPC通信更加高效（例如，比网络的必要传输时间短5倍）。 这似乎很重要，以免通信变得如此昂贵，以至于应用程序设计人员会竭力避免这种情况。 否则可能会开发的应用程序将因其避免通信的愿望而失真。 另外，我们认为在不损失简单性或效率的情况下，使RPC包的语义尽可能强大很重要。 否则，将要求应用程序程序员在RPC包之上构建额外的机制，从而失去单一统一通信范式的收益。 设计中的一个重要问题是解决强大的语义和效率之间的矛盾。

我们最终的主要目标是提供与RPC的安全通信。 以前实现的协议都没有任何条款可以保护我们网络上传输的数据。 即使是密码也以明文形式传输。 我们的信念是，对通过开放网络进行安全通信的协议和机制的研究已经到了一个阶段，在这个阶段中，对于我们来说，将这种保护包含在我们的软件包中是合理且可取的。 另外，很少（如果有的话）分布式系统以前提供过安全的端到端通信，并且从未应用于RPC，因此该设计可能会提供有用的研究见解。