特别放送 | 谈谈我所经历过的RPC

2020-04-17 何小锋

RPC实战与核心原理 进入课程 >

讲述：张浩

时长 09:00 大小 8.26M

你好，我是何小锋。上一讲我们学习了如何在线上环境里兼容多种 RPC 协议，目的就是为了能够平滑地升级线上环境中已经存在的 RPC 框架，同时我们也可以利用多协议的特点来支持不同的使用场景。

以上就是我们整个专栏关于技术内容的最后一讲了，很幸运能够和你一起携手并肩走过这些日子，这段时间我们跨了新年，也经历了让人猝不及防的新冠肺炎。现在我才发现原来能够自由地呼吸新鲜空气也是一种幸福，祝你平平安安度过这段艰难的日子，期待我们能早日摘下口罩。为了感谢你这些日子的陪伴，今天我们来换换脑子，聊一些轻松的话题。我来分享分享我所经历过的 RPC 框架吧。

与 RPC 结缘

我 1998 从大学毕业的时候，互联网并不是像今天这样如火如荼地进行着。那时候大部分 IT 公司都在耕耘数字化办公相关领域，所以大学毕业的时候我也就随了大流，进入了一个从事办公软件开发的公司，也就是我们今天经常说的“传统软件行业”。

后来随着互联网的快速普及，各个行业都想借着这个机会跟互联网发生点关系，因为大家都知道互联网代表着未来，在咱国内借助互联网技术发展最快的行业代表就是电商和游戏。

得益于互联网的普及，各个互联网公司有关技术的文章也是扑面而来，比如 A 公司因为流量激增而导致整站长时间瘫痪，B 公司因为订单持续上涨而导致数据库压力太大等等。每当我看到类似技术文章的时候，我都在想自己什么时候也可以遇到能用技术优化的方式来解决的问题呢。后来发现这些“设想”在我当时所在的行业里很难体会到，所以 2011 年我毅然选择了加入京东。

那时候我们公司刚好处于一个业务高速发展期，但系统稳定性相比今天来讲的话，还处于一个比较稚嫩的阶段。在日常工作中，我们的研发人员不仅要面对来自业务方需求进度的压力，还要面对因为线上业务增长而导致的系统压力。这样的双重压力对我来说很新鲜，这种体验是我之前从未有过的，这让我感到很兴奋。

之前我自己也做了很长一段时间的业务系统开发，虽然积累了不少实用的编程技巧，也认真阅读过国内外很多优秀的开源代码，包括像 Spring、Netty 等等这样优秀的框架，但因为从来没有经历过这么多系统之间复杂调用的场景，所以在入职后的一段时间里，我的内心一直处于忐忑状态。

得幸于同事的热情帮助，我很快地就融入了团队氛围。因为我们大团队是负责整个公司基础架构的，可以简单理解成我们就是负责公司 PASS 平台的建设，既然是 PASS 平台那肯定少不了中间件，而中间件里面最核心的应该就是 RPC 了，因为 RPC 相对其它中间件来说，使用频率是最高的，也是我们构建分布式系统的基石。

从此我就踏上了 RPC 这条“不归路”了，当然并不是因为这是条绝路，而是这条路一直充满着挑战，并没有让我觉得有停下来的想法。

使用过的 RPC

因为我工作年限比较长，所以编程经历相对也比较丰富，在转 Java 之前我做过一段时间的.Net。早期因为主要做数字化办公软件，所以那时候用.Net 编程的机会比较多，主要是

因为很多应用采用.Net 相对使用 Java 来说，开发效率会高很多。但后面应用复杂之后，.Net 在性能、可维护性上都不如 Java 有优势，而且.Net 在很长一段时间内并不支持在 Linux 环境下部署，所以我们就把应用都逐渐改成 Java 了。

ICE

这就存在一个问题，我们是怎么把.Net 应用平滑地切换到 Java 上呢？原本是.Net 应用之间的互相调用，需要改成.Net 跟 Java 之间互相通信。为了解决这个问题，我们当时选择了一个比较古老的 RPC 框架 ICE（https://zeroc.com），可能现在很多人都没有听说过了。

Hessian

后面使用 Java 开发应用的机会越来越多，而且随着 Spring 开发方式的大流行，在 Java 应用里面使用 ICE 来完成应用之间的 RPC 调用就变得比较鸡肋了。所以当时我们用了一种新的 RPC 框架 Hessian（http://hessian.caucho.com/），这时我们就把 Java 应用之间的 RPC 调用方式改成了 Hessian 方式。

用 Hessian 的原因就是因为它可以很好地跟 Spring 进行集成，对 Spring 项目的开发人员来说，开发一个 RPC 接口就变得很容易了，我们可以直接把 Java 类对外进行暴露，用作 RPC 接口的定义，而不需要像 ICE 一样先定义 Stub。

单纯的从 RPC 框架角度出发，Hessian 是一款很优秀的产品，即使放到今天，它的性能和鲁棒性都有着很强的参考意义。但当业务发展壮大到一定程度后，应用之间的调用就不仅仅需要考虑用什么 RPC 框架了，更多的是需要考虑怎么去完成服务治理。

另外一个原因就是因为 Hessian 是没有服务发现功能的，我们只能通过 VIP 暴露的方式完成调用，我们需要给每个应用分配一个 VIP，把同一接口的所有服务提供方实例挂载到同一个 VIP 上。这种集中式流量转发架构就会使得提供 VIP 服务的 LVS 存在很大的压力，而且集中式流量的转发会让调用方响应时间相对变长。

Dubbo

为了解决类似 Hessian 这种集中式问题，实现大规模应用服务化的落地，国内的 RPC 框架 Dubbo 的做法就显得比较先进了。随着业务越来越复杂，应用之间的调用关系也就变得更加错综复杂，所以后面我们也是选择基于 Dubbo 进行扩展，以完成 RPC 通信，而服务发

现则通过接入 Zookeeper 集群来完成。通过这种现有框架的搭配，我们完成了应用服务化的快速落地，也同时完成了统一公司内部所有应用 RPC 框架的目标。

但随着微服务理念越来越流行，很多应用的接口也是越拆越细，导致我们 Zookeeper 集群需要接入的接口数量越来越多；还有就是因为我们每年的业务量是成倍增长，为了让应用能够抗足够的调用量应用，我们也需要经常扩容，从而导致 Zookeeper 集群接入的 IP 实例数也是呈数量级增长的，这使得我们的 Zookeeper 集群负荷特别重。

再有就是 Dubbo 相对有点复杂，而且性能还有提高空间，这使得我们不得不考虑新的方案。

自研 RPC

在这种背景下，我们决定自行研发一套适合自己业务场景的微服务解决方案，包括 RPC 框架、服务治理以及多语言解决方案。

至此我们自研的 RPC 就一直平稳地支持着公司内的各种业务。

这几年，在以 K8S 为代表的基础设施演进过程中，一个重要的关键词就是应用基础设施能力的下沉。在过去我们给应用提供 RPC 能力的时候，都是需要应用引入 Jar 包方式来解决的，在 RPC 里面，我们要把服务发现、路由等一整套 RPC 解决方案都融入到这个 Jar 里面去。

未来

目前，K8S 已成为基础设施的事实标准。而原先通过 Jar 包的方式封装的各种基础设施能力，现在全都被 K8S 项目从应用层拽到了基础设施中。那对于我们 RPC 来说也是一样，我们需要把非业务功能从传统的 RPC 框架中剥离出来，下沉到基础设施并且融入基础设施，然后通过 Mesh 去连接应用和基础设施。

这也是 RPC 发展的下一个阶段，完成所有应用的 Mesh 化。所以说，RPC 这条路没有尽头，只有不断的挑战和乐趣。希望你也能爱上它！

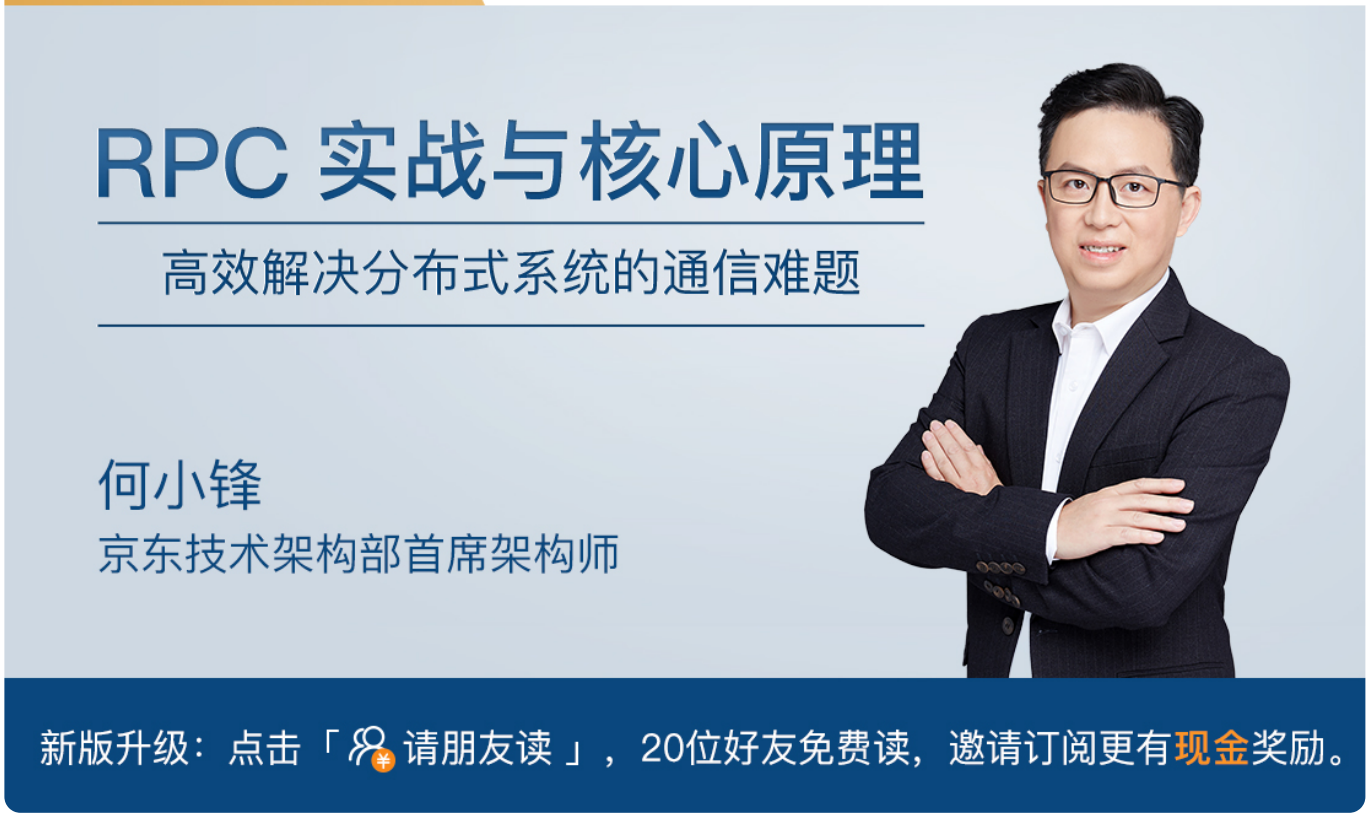
那最后我还想和你谈谈我对 RPC 的看法。

可能大家在谈论 RPC 时候，都想着 RPC 只是解决应用之间调用的工具。从本质上来讲，这没有什么问题，但在现实中，我们需要 RPC 解决更多的实际问题，比如服务治理，这些东西都是在使用 RPC 的过程中需要考虑的问题，所以我个人认为 RPC 应该是一个比较泛的概念。

当然，可能我们中大多数人现在是没有机会去完整实现一个新的 RPC 的，这不仅是精力的问题，更多是实际需求的问题，那为什么我们还需要学好 RPC 呢？我的想法很简单也非常实在，就是因为 RPC 是我们构建分布式系统的基石，就好比我们每次都是从“Hello World”开始学习一门新的编程语言。期待你能打牢这个基础，总有一天你会体验到它的能量！

今天的特别放送就到这里，也非常期待听到你和 RPC 的故事。

更多课程推荐



新版升级：点击「请朋友读」，20位好友免费读，邀请订阅更有现金奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇　24 | 如何在线上环境里兼容多种RPC协议?

下一篇　结束语 | 学会从优秀项目的源代码中挖掘知识

精选留言 (9)

写留言

xiaoa
2020-04-17

期待老师更多点心路的分享，感谢老师分享

展开

作者回复: 希望对你有帮助

1

Reason
2020-04-17

非常感谢老师的分享。最感兴趣的一点是，非业务功能从传统的 RPC 框架中剥离出来，下沉到基础设施并且融入基础设施，能否有幸听到老师关于这一点的具体展开呢?

作者回复: 最明显的就是服务发现

1

忆水寒
2020-04-21

感想颇多，在不同业务场景下选择不同的RPC框架，我们自己也实现了一套自研的RPC，可以切磋切磋。

问心
2020-04-21

老师，上了k8s以后，rpc内部的路由、权重、负载均衡等组件是不是就不需要了?

zgscy100
2020-04-19

非常感谢老师的付出，带来这么好的系列文章。每篇文章后都提有一个问题，老师能不能再筛选出一些问题，给出一个详细权威的解答呢

展开

作者回复: 已经写了一篇了

夜空中最亮的星（华仔...

2020-04-17

多来几篇这样的分享心路的文章吧

展开

上帝之手

2020-04-17

老师，您好，我很好奇，你们选择了dubbo，却没有选择spring cloud，虽然我没有研究过dubbo，但看过spring cloud核心代码的一部分，我觉得大体上跟您描述的RPC实现的功能上来说差不多。网上很多资料说spring cloud也要强于dubbo，是不是因为spring cloud是基于http的，性能不够高的原因？还有一个疑问，spring cloud它属于RPC框架吗，网上资料也是层出不穷，各持己见。谢谢老师

展开

作者回复: 不同时期吧，至于dubbo和Spring cloud选型可以根据实际需要。

1

凌霄

2020-04-17

多次听到鲁棒性，看来鲁棒性大于性能

展开

作者回复: 越是基础设施，稳定性肯定第一优先

Jackey

2020-04-17

感谢老师的分享，第一遍阅读收获很大，进阶篇的内容感觉难度还挺大的，打算二刷

作者回复: 加油