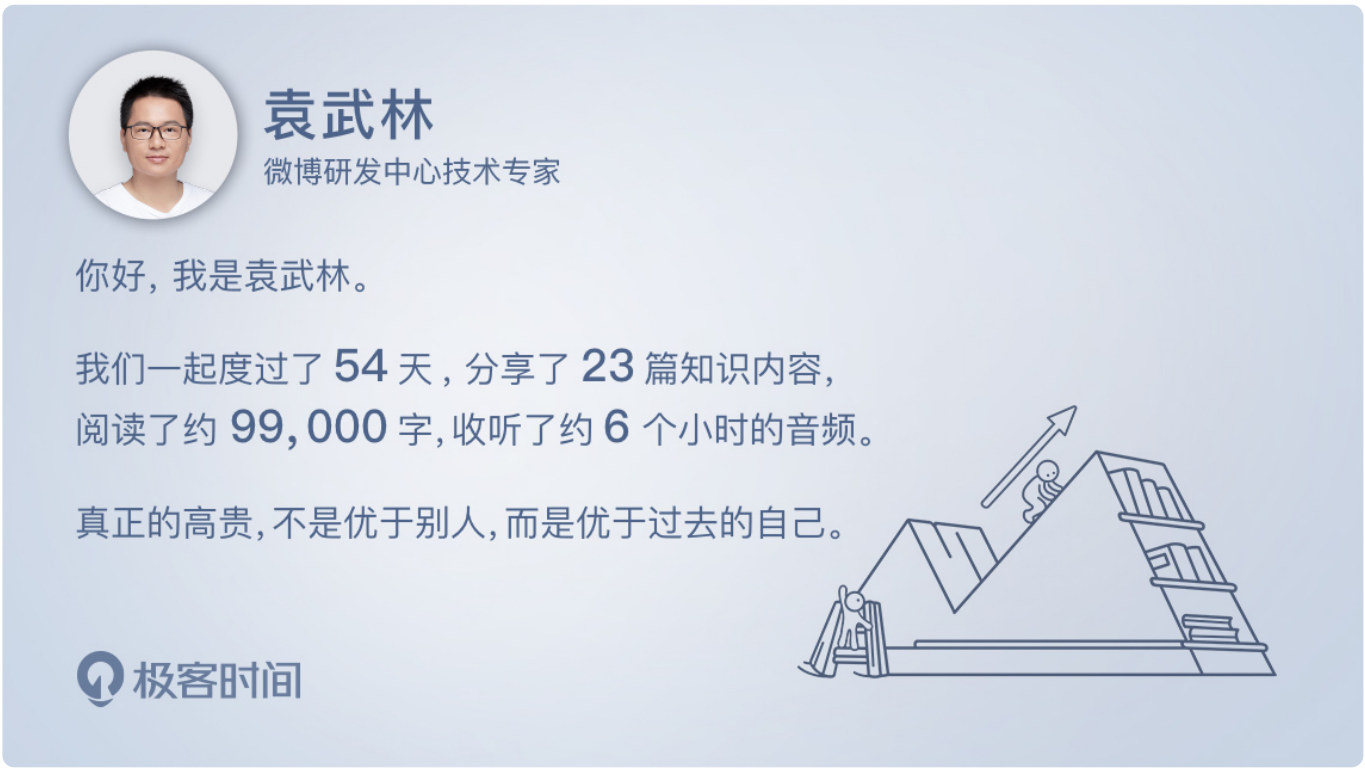
# 结束语 | 真正的高贵，不是优于别人，而是优于过去的自己

2019-10-18 袁武林

即时消息技术剖析与实战 进入课程 >




**讲述：袁武林**
时长 08:48 大小 7.06M

你好，我是袁武林。

不知不觉中，在大家的耐心陪伴下，即时消息专栏的更新马上就要结束了。

首先，要感谢各位同学对我的信任和宽容，给我这个机会，让我能够在这段短暂的时间里，带领大家一起来学习即时消息这一门古老但又充满活力的技术。

对我来说，这是一段非常特别和有趣的体验，我也第一次尝试通过专栏的方式，将自己了解的知识进行体系化的输出。

当然这并不是一件简单的事情，知识点整理、稿件打磨、录音、留言反馈等等，背后的艰辛可能也只有经历过的人才能体会到。但每次专栏更新上线，看到自己的付出有帮助到一些同

学，并且得到肯定的时候，我瞬间感觉又充满了战斗的能量。

## 关于专栏的落地

实际上，在开始筹备这个专栏的时候，我个人是比较没有信心的。

原因倒不是说技术层面的问题，而是担心即时消息这个话题，因为受众相对没有那么广泛，在推出之后，如何让更多非 IM 行业的小伙伴，也能够感兴趣并且参与进来，一直是我和极客时间团队需要认真考虑的一个事情。

因为一直没有想到更好的办法，所以这个专栏甚至中途一度停滞中断。直到 6 月下旬，经过和极客时间小伙伴们的多次讨论，我们才最终决定继续推进这个专栏的落地。

## 关于即时消息技术的前景

从我们的角度看，随着 4G 网络的普及和 5G 网络的逐步推广，在移动网络场景中，不管是流量资费还是网络的稳定性和速度、带宽，相比几年前都有了非常明显的提升。

在这样的大环境下，原本更多被用于聊天等社交场景的即时通讯技术，也被大范围应用于“万物互联”的物联网场景，以及新型社交模式，如直播互动、游戏互动等。而且，还有越来越多对实时性要求高的业务场景，也都开始引入即时通讯技术，来提升用户的使用体验。

因此，我们觉得，开设一门即时通讯技术相关的课程，不仅能够让同学们从基础原理层面上更体系化地了解即时通讯底层核心的技术实现，并且能够帮助大家在各自从事的业务系统中，结合实际场景，来尝试引入课程中介绍到的某些适合自己业务的技术点。

专栏真正开始落地后，在一系列自我摧残式的存稿，以及逐字逐句和编辑的磨稿过程中，我才切身体会到做一个专栏有多不容易。

和以前随意写写博客相比，一个严谨的专栏需要耗费大量的时间和精力去提炼技术点，并且要尽量用“通俗易懂”的文字来组织语言。

专栏的落地对于我自己来说，也是一个成长和学习的过程。

特别是在留言区，就某些技术点和各位同学们的讨论，大家的一些有趣、富有想象力的观点，也能让我重新思考，涉及的这些技术点是不是某些方面的考虑还不够，或者是否还有更好的实现落地方案。

在专栏即将结束的时候，我也想借这个机会，和参与到这门课程学习中的你，聊一聊我对技术和学习方法的一些个人的思考。

## 学习方法论：知识的广度和深度问题

很多时候，我们搞技术的小伙伴经常会听到人们对于知识的两种学习路径：一种是追求知识的广度，力图多点开花，前后端通吃；另一种是追求某一个知识点的深度，从对某一个小知识点的“了解和会用”，到“底层原理分析”，再到“思考优化”的这种路径来学习知识。

但是对于很多从事技术的小伙伴们来说，这样很容易陷入到一个“两难”的学习方法论的抉择中，毕竟现在的新技术层出不穷。

就拿即时消息的技术体系来说，暂且不去讲多种语言实现的问题，光是 Java 体系下网络通信的 NIO 框架就有好几种，要选择哪个框架就是一个让人很纠结的问题。

这种情况下，很容易让人陷入一个怪圈：就是我们可能花费了大量的精力，放在框架特性介绍对比、框架使用方法研究等知识上面，力求通过掌握更多的 NIO 框架的使用，来提升自己这方面知识的广度。但从我个人角度来看，是不推荐这种学习和研究方式的。

我认为所谓的知识广度是一个伪命题，换句话说，知识的广度只是一个结果，而不是一种学习的方法。

我更推荐的是：**针对某一个特定的技术点，从使用方法下沉到原理层面上，再逐步去拓展和了解研究学习过程中碰到的各种疑问与实现细节，从而“由点到面”地去吃透该知识点。**

打个比方，要对 NIO 框架进行研究，我觉得没必要先去纠结各种框架的优劣，而是可以选择一种比较主流大众的框架去做实践；然后，逐步去了解这个 NIO 框架使用到的 JDK 的 IO 库的实现方式。

比如，搞明白 JDK 的 IO 库模型，是从传统的阻塞型的 BIO，发展到 JDK 1.4 开始支持多路复用的 NIO，再发展到 JDK 1.7 继续改进的异步的 AIO，以此来了解这几种模型的区别和迭代改进的地方。

最后，再深入到底层的操作系统层，了解它们是如何支持这些 IO 模型的。

比如，要了解 Linux 的 select 和 epoll 的实现原理，可能还需要了解 IO 过程中涉及到的 IO 事件、最终的 TCP 连接的状态，以及与数据收发之间的关系。因此，你还可以对 TCP 协议进行体系化的了解。

通过这种层层深入的学习方式，我们就能较为扎实地掌握一个 NIO 框架涉及到的垂直知识体系。

这样，你在掌握知识点的同时，知识广度方面也能自然而然地充实起来，以后再切换到其他 NIO 框架，就相对简单了很多，也能更客观和精确地对这些框架从底层原理层面来进行比对了。

当然，这种逐层深入的学习方法，在前期可能需要花费大量的精力和时间，因此需要你一定的耐心和坚持，但在对底层原理层知识有一定的积累之后，这种学习的方法也会越来越轻松。

## 学习方法论：碎片化知识和系统化学习

关于技术学习路径的另一个小小的建议是：从了解碎片化的知识开始，逐步扩大你在某一方面的技能树，然后再对这方面的整体知识进行系统化的学习和总结沉淀。

随着现在各种技术博客、技术类公众号、技术站点等等的普及和传播，大部分时候，我们只能通过工作之余的些许碎片化时间，来了解某些细碎的知识点。

当然，这些碎片化的知识点是一种不错的积累，但这些零散的知识点，在碎片化学习的时候，由于学习时间短、文章内容比较聚焦于一个小点，很容易让我们看完之后产生一种“我知道了”，然后就没有后续了，难以进行复杂的思考。

所以，我个人的建议是，**对于某一技术点方面的碎片化的知识，在一段时间的学习后，还是需要进行系统化的“复盘”**，一来可以补充遗漏的某些技术细节，二来能够形成对这个知识点的整理和总结思考，最终沉淀成自己的文档或者代码输出。

另外， 你也可以尽量优先地去选择学习与自己工作内容息息相关的技术点，这样在理解知识和后续的输出实践上，也会更有帮助。

## 写在最后

对于从事技术工作的我们来说，技术探索的道路总是显得如此漫长，不断迭代更新的技术时常会让我们眼花缭乱，各种技术知识产品的爆炸式推广，让我们获取知识的方式变得如此简单。

但另一方面，在快节奏的生活方式下，在繁重的工作、与家人朋友的相处时间之外，我们的时间总是显得远远不够用。

如何保持一颗对新鲜事物的好奇心，如何高效有序地管理时间，如何形成一套适合自己的学习方法，这个时候可能往往比技术本身更重要。

海明威说：**“真正的高贵，不是优于别人，而是优于过去的自己。”**

无论如何，可能我们在一开始的时候，学习的技巧并不是那么高明，**但只要我们保持终身学习的姿态，持续地投入到真正喜欢的事情上，就能让今天的你比昨天进步一点点。**

**我相信，时间的复利最终能让你真正采撷到成功的芬芳玫瑰！**

上一篇　22 | 答疑解惑：不同即时消息场景下架构实现上的异同

## 精选留言 (7)

写留言

**leslie**
2019-10-18

非常喜欢陈皓老师在二叉树视频中的一句话“芝兰生于空谷，不以无人而不芳”。学到现在听过各个老师的方式：最终还是选了刘超老师的方式改进成自己的。

学习其实很简单：从开课一路坚持到最后一节课一周不拉的学完，其实学习的过程中还是碰到了大量的问题，导致不得不去扩展学习，不过庆幸的是最后坚持的学完了；坚持到最后其实就是一种胜利。学习的过程其实就是不断的自我否定自我前进的过程：定期...

展开

作者回复: 开源的监控系统很多呀像我们现在就在用graphite，grafana，zipkin这些，也确实不错，有机会可以分享下我们在这一块的工程实践。另外，也谢谢你对我的专栏的支持，有问题可以随时交流。

2

**墙角儿的花**
2019-10-18

感谢老师，很受益。终究要再见，不能再进行交流了，好遗憾

作者回复: 感谢你的支持，不说再见，可以随时在留言区和我交流的哈。
另外也可以关注
我的微博：https://weibo.com/u/1243432494
我的个人博客：https://coldwalker.com

最后借留言区打个广告，我们团队在招聘IM高级工程师，有兴趣的同学欢迎应聘这个职位:
https://www.lagou.com/jobs/6487085.html

1

**小可**

2019-10-18

这个专栏是我追的比较紧的课程，之前虽然我没做过IM系统，但对IM有点兴趣，网上虽然有类似的博客文章，但都不是很深入。我想知道真正的IM系统是怎么实现的，是不是和我想的一样？一节一节看着专栏，也在一步一步验证我的想法。有些点，经过老师的讲解也是茅塞顿开，真的感谢袁老师真心细致的讲解，谢谢。

展开

作者回复: 感谢支持和肯定，你们能够真正从中学到东西就是我最大的收获。也相信你的努力会得到应有的回报。

1

**clip**

2019-10-18

说来惭愧，之前买了很多的专栏但完成度都比较低。这个专栏是第一个几乎跟着更新看完的，留言的参与度也是最高的。因为现在的工作是 IM 相关的业务开发，但是因为实际开发接触面有限不知道整体的架构。学习专栏之后通过和已经知道的一些信息做对比就有更深的认识了，对之后的工作也会很有帮助。多谢大佬的专栏！

展开

作者回复: 谢谢你的肯定，能够对你们有所帮助是我最开心的事情。

1

2019-10-18

真正的高贵，不是优于别人，而是优于过去的自己。
完整学了一遍，很不错的专栏，还会二刷三刷，袁老师辛苦了。

展开

作者回复: 谢谢你的肯定，与你共勉。有问题可以留言区随时再交流。

1

**云**

2019-10-18

老师好，老师的专栏结束了，感觉还没有学够，感觉要学的还太多。如何把im落地生产环境还是摸不着头脑，好在老师已经给我们提供了一个demo。以前没接触过im，问您一个问题。我们想在app中做个消息提醒功能，您说我们该用什么技术，用户在2万左右。目前

想用websocket做可以吗？还是用http2的推送。或者在前端做轮询定时请求后端接口就好。还有哪些好的方案？我们用的springboot。

展开

作者回复: 用户量小的话可以先尝试用轮询来实现，这种实现成本是最低的，然后再逐步发展的话可以基于mqtt来做长连推送，或者直接用第三方的推送服务，用户量小的话也花不了多少钱。

1 1

**我来也**

2019-10-18

老师辛苦了！
现在im确实很流行，slack实在是太厉害了。
恰好最近的项目也是作im的，就再看看别人是怎么做的。等到后期用户量上去了，也知道该往什么方向优化了。

展开