

结束语 | 以梦为马，莫负韶华！

胡夕 · Kafka核心技术与实战



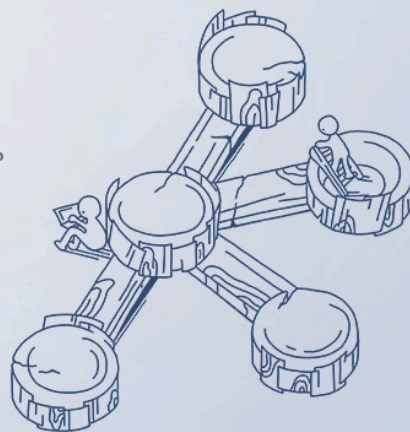
胡夕

人人贷计算平台部总监，Apache Kafka Contributor

你好，我是胡夕。

我们一起度过了 **102** 天，共学习了 **45** 篇文章，
阅读了 **163,558** 字，收听了约 **10** 个小时的音频。

以梦为马，莫负韶华！



你好，我是胡夕。今天，我的专栏“Kafka 核心技术与实战”就正式结束了。

回顾与你在专栏相聚的这几个月，我的内心充满了成就感。且不必说这 42 讲的文字全是我一字一键敲下来的，也不必说那长达十几个小时的录音，单是留言区那些踊跃积极的提问与讨论，就足以使我深受感动并收获满满了。

此时此刻，千言万语汇成一句话：“感谢！”感谢你对我和本专栏的支持，感谢你曾经的鼓励与提问，也感谢你的肯定与期望。另外，我也要向你表示祝贺，祝贺你完整地学习了专栏的全部课程，你的恒心与坚持令人敬佩。

虽然专栏更新结束了，但是我相信我们的 Kafka 学习之旅不会结束。相反，这对于你来说，或许是一个新的开始。

还记得开篇词里的那句话吧：“**Stay focused and work hard!**”我一直觉得，学习任何技术，甚至是搞定任何事情，只要下足了功夫，理论上你可以藐视一切学习方法或捷径。但是，

如果你忽视了毅力和坚持，再多的速成教程也无法引领你达到你期望的高度。著名的“10000 小时定律”就明确表示，**10000 个小时的锤炼，是所有人从平凡人变成世界级大师的必要条件。**

还是那句话，只要你持之以恒地投入时间去学习，你就能成为某个领域的专家。因此，从某种意义上说，我这碗“鸡汤”的配料非常简单，就四个字：**干就完了。**

那这是不是在说书籍、专栏之类的他人智慧总结就没用了呢？当然不是！他山之石，可以攻玉，书籍和专栏的最大作用就在于，当你遇到岔路口时，它们能够帮助你快速地识别前进中的已知路障，让你少走弯路，更快地实现目标。但**前提是你要在路上，而不是单纯地想要依赖它们速成。**

在专栏的最后，我想再和你分享一些学习大数据框架的个人经验。这些经验不仅仅适用于学习 Kafka，对于其他框架甚至是分布式系统的学习，都是适用的。

首先，最重要的就是夯实技术基本功。这是我们 IT 从业者赖以生存的基石。

这里的基本功包含很多方面，比如**操作系统、数据结构**等，但我更想说的，还是对 **Java 语言的掌握。**

目前，大数据框架多是以 Java 或 JVM 系语言开发而成的，因此，**熟练掌握甚至精通 Java，是学好大数据框架的基石！**所谓精通，不仅仅是要求你熟练使用 Java 进行代码开发，更要求你对 JVM 底层有详细的了解。就这个层面的学习而言，我想给你 3 条建议。

1. **持续精进自己的 Java 功底。**比如，你可以去 Java 官网上，把 Java 语言规范和 JVM 规范熟读一遍。很多人都不太重视语言规范文档，但实际上，Java 中关于线程和同步的知识，在 Java 语言规范中都有相关的阐释。
2. **提升自己的 Java 多线程开发以及 I/O 开发能力。**很多大数据框架底层都大量使用 Java 多线程能力以及 NIO 帮助实现自身功能。就拿 Kafka 来说，多线程自不必说，Kafka 可是大量使用 NIO 实现网络通信的。所以，这部分的知识是你必须要熟练掌握的。
3. **掌握 JVM 调优和 GC。**我推荐你去读一读“Java Performance”这本书。虽然目前 GC 收集器大部分演进到了 G1 时代，但书中大部分的调优内容依然是适用的。调优 Kafka 的

JVM，也要依赖这部分知识给予我们指导。

除此之外，你还要学习分布式系统的设计。

分布式系统领域内的诸多经典问题，是设计并开发任何一款分布式系统都要面临和解决的问题，比如我们耳熟能详的一致性问题、领导者选举问题、分区备份问题等。这些问题在 Kafka 中都有体现，我们在专栏里面也有所涉及。因此，**分布式系统的诸多基础性概念，是帮助你日后深入掌握大数据分布式框架的重要因素。**

而且，很多经典的分布式问题在业界早已被研究多年，无论是理论还是实践案例，都有着翔实的记录。比如我们在专栏前面谈到的分区概念，分区在分布式系统设计中早就不是什么新鲜的概念了，早在上世纪六七十年代，就已经有行业专家在研究分区数据库的实现问题了。要较好地掌握大数据框架中的分区或分片，是不可能绕过分布式系统中的分区以及分区机制的。

这些经验都偏重理论的学习。你千万不要小看理论的价值，毕竟，列宁说过：“没有革命的理论，就没有革命的运动。”这里的“运动”就是一种实践。先让理论指导实践，再借助实践补充理论，才是学习任何东西无往而不利最佳法则。

强调完理论，自然就要引出实践了。我这里所说的实践不仅仅是对框架的简单使用。你从官网上下载 Kafka，启动它，然后创建一个生产者和一个消费者，跑通端到端的消息发送，这不叫实践，这只是应用罢了。**真正的实践一定要包含你自己的思考和验证，而且要与真实业务相绑定。**我不排斥你单纯地学习某个框架，但以我个人的经验而言，在实际工作中进行学习，往往是学得最快、掌握得也最扎实的学习方式。

另外，在实际学习过程中，你最好记录下遇到问题、解决问题的点点滴滴，并不断积累。要知道，很多技术大家之所以成为技术大家，不仅仅是因为理论掌握得很牢固，填过的“坑”多，更重要的是，**他们不重复犯错。**

孔子曾经称赞他的学生颜回“不贰过”。“不贰过”也就是不重复犯错。在我看来，在实践方面，一个不犯相同过错的人，就已经可以被称为大家了。

好了，这些就是我在学习技术的过程中总结的经验，现在全部分享给你，希望对你有所帮助。

我为你准备了一份结课问卷，题目不多，两三分钟就可以完成。希望你能畅所欲言，把自己真实的学习感受和意见表达出来，我一定会认真看，期待你的反馈。



胡夕

人人贷计算平台部总监, Apache Kafka Contributor



不知道在学习过程中，你有哪些体会和评价？
这里有一份专栏调查问卷，邀请你填写。

**在9月20日前提交，
极客时间赠送给你专属优惠券。**

我们一起继续成长！

去提交

最后一期了，也欢迎你在这里给自己的专栏学习做个总结。关于专栏或是其他内容，如果你还有什么问题，也尽管提出来，我会持续回复你的留言。

送君千里，终须一别。所谓行百里者半九十，愿你继续在技术这条道路上深耕，并得偿所愿，不负韶华。我们江湖再见！

经验分享

- 最重要的就是夯实技术基本功。熟练掌握甚至精通Java，是学好大数据框架的基石！跟你分享3条建议：持续精进自己的Java功底；提升自己的Java多线程开发以及I/O开发能力；掌握JVM调优和GC。
- 要注重学习分布式系统的设计。
- 真正的实践一定要包含你自己的思考和验证，而且要与真实业务相绑定。
- 在实际工作过程中，最好记录下遇到问题、解决问题的点点滴滴，并不断积累。
- 不重复犯错。



极客时间

AI智能总结

这篇文章是一篇技术专栏的结束语，作者总结了自己的专栏学习经验，并向读者提供了学习大数据框架的个人经验。作者强调了夯实技术基本功的重要性，特别是对Java语言的掌握。他还提到了学习分布式系统设计的重要性，并强调了理论学习和实践相结合的学习方式。作者鼓励读者持之以恒地投入时间去学习，相信只要持续

努力，就能成为某个领域的专家。最后，作者邀请读者填写结课问卷，并欢迎读者在专栏学习做个总结，提出问题。整篇文章充满了作者对读者的鼓励和期待，为读者提供了宝贵的学习经验和建议。

© 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。

全部留言 (52)

最新 精选



张丽娜

2020-03-04

胡老师您好，专栏已经结束了，我目前还在刚刚开始学习。因为工作中同事在用，所以想要深入了解一下这块的知识。可专栏偏重于Kafka原理相关的介绍，自工作后我就没有使用过JAVA语言，所以感觉学习这门理论课程时，感觉很多地方都是似懂非懂，且日常工作中也不会用到，所以理解起来就更抽象了。想要请教下老师，针对于我这种情况的话，老师能否给些学习的建议？还是先硬肯一遍专栏，接着学习JAVA语言？还请老师能够有空的时候，指点一二，提前感谢老师！

作者回复: 首先学习Kafka并没有要求必须有特别强的Java功底。主要还是取决于你后续使用哪种语言的Kafka客户端进行开发。比如很多人使用C++来开发Kafka应用程序。我建议先把官方文档好好看一遍，然后再结合专栏里面的知识点查漏补缺。



👍 2



@%初%@

2020-11-11

终于完成了专栏的学习，对kafka整体的轮廓，设计有了清晰的认识，结合老师推荐的《深入理解kafka技术》继续学习。收获满满呀。

作者回复: 感谢鼓励：)



👍 1



滚去滚来

2020-05-21

谢谢老师！整个专栏让我收获满满，项目中需要，准备在实践中学习

作者回复: 加油👊



sunny

2020-01-17

个人感觉是市面上最好的专栏，学习了！！！！

作者回复: 感谢鼓励：)



保密!

2021-09-08

山高路远，定会江湖再相见！

作者回复: 加油！



邈邈的流浪剑客

2021-07-07

最近复习Kafka的文章，大家感兴趣可以一起来学习哈：

https://blog.csdn.net/qq_40378034/article/details/118280352

https://blog.csdn.net/qq_40378034/article/details/118538632

作者回复: 强



芝士烤松饼

2021-02-21

我也想像你一样强，我会努力的！

作者回复: 加油！





杨阳

2020-11-04

打卡完结

作者回复: 加油!



Cooper

2020-10-25

thanks

作者回复: no problem:)



风轻扬

2020-07-13

老师, 请教一个问题。kafka版本: kafka_2.10-0.10.0.0, zk用kafka自带的, 单分区。某一天晚上, kafka的server.log里面一直报一段错误。

Preparing to restabilize group Idm-Kafka-Syn-Consumer with old generation 0 (kafka.coordinator.GroupCoordinator)

[2020-07-13 17:43:39,829] INFO [GroupCoordinator 0]: Stabilized group Idm-Kafka-Syn-Consumer generation 1 (kafka.coordinator.GroupCoordinator)

[2020-07-13 17:43:39,830] INFO [GroupCoordinator 0]: Assignment received from leader for group Idm-Kafka-Syn-Consumer for generation 1 (kafka.coordinator.GroupCoordinator)

[2020-07-13 17:44:09,831] INFO [GroupCoordinator 0]: Preparing to restabilize group Idm-Kafka-Syn-Consumer with old generation 1 (kafka.coordinator.GroupCoordinator)

[2020-07-13 17:44:09,832] INFO [GroupCoordinator 0]: Group Idm-Kafka-Syn-Consumer generation 1 is dead and removed (kafka.coordinator.GroupCoordinator)

[2020-07-13 17:45:06,817] INFO [Group Metadata Manager on Broker 0]: Removed 0 expired offsets in 0 milliseconds. (kafka.coordinator.GroupMetadataManager)

[2020-07-13 17:55:06,817] INFO [Group Metadata Manager on Broker 0]: Removed 0 expired offsets in 0 milliseconds. (kafka.coordinator.GroupMetadataManager)

我在kafka的jira里面看到了一个提交, <https://issues.apache.org/jira/browse/KAFKA-4051>。kafka服务端正在进行rebalance时, 如果更改服务器时间会导致rebalance无法结束, 可能会导致以上问题。然后, 我就重新下载了相同版本的kafka。重新部署启动, 该问题并未解决。

zookeeper的data目录，我没有删除，还是用的之前的。消费者参数如下：

```
enable.auto.commit" value="true"
auto.commit.interval.ms" value="5000"
zookeeper.session.timeout.ms" value="55000"
rebalance.max.retries" value="12"
rebalance.backoff.ms" value="6000"
key.deserializer"
    value="org.apache.kafka.common.serialization.StringDeserializer"
value.deserializer"
    value="org.apache.kafka.common.serialization.StringDeserializer"
```

其他都是默认

老师帮忙看一下，是什么原因？

作者回复: 你是说修改了服务器的系统时间吗？如果确实能够稳定地复现，那么有可能是一个bug。可以提交给社区。上面都是INFO，没有看到错误日志，不太好进一步判断

