



下载APP

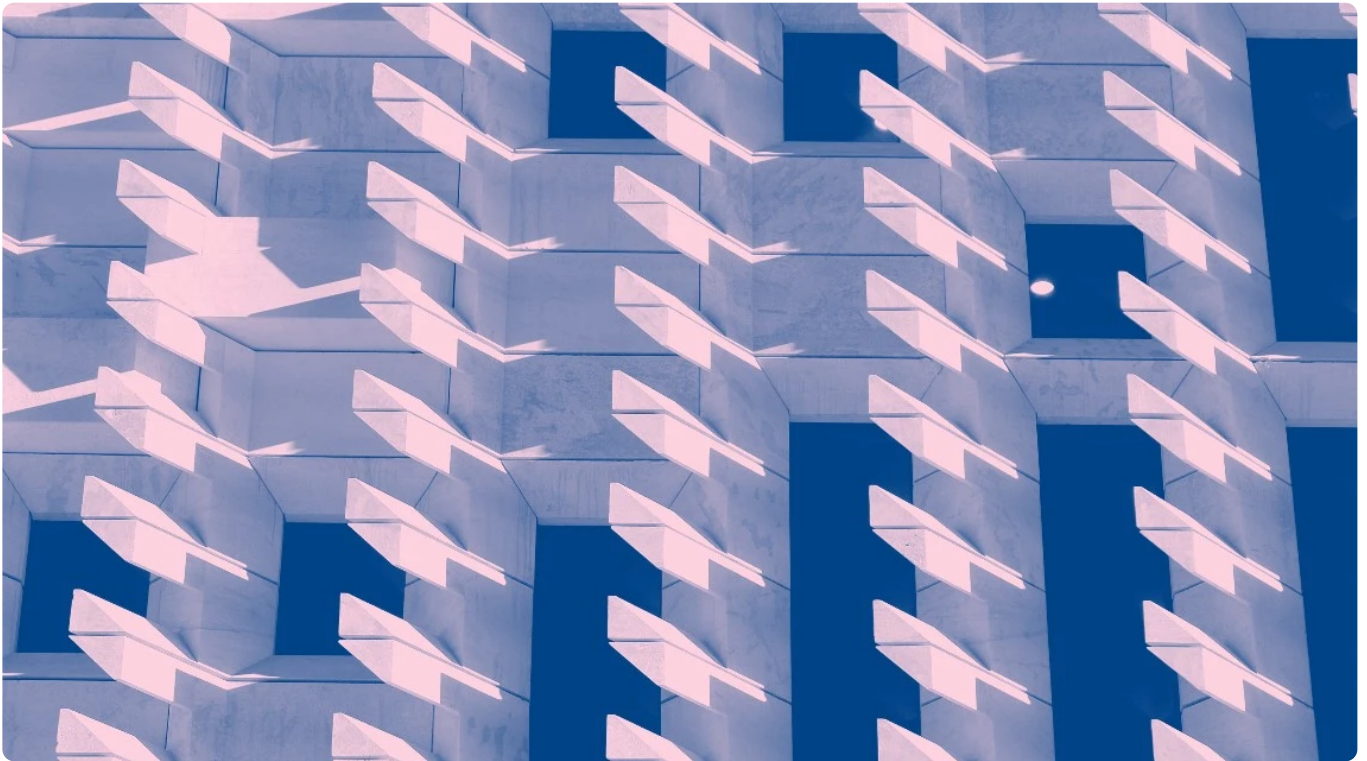


用户故事 | 黄涛：水滴石穿、坚持不懈，必能有所精进

2022-01-12 黄涛

《大数据经典论文解读》

课程介绍 >

**讲述：王惠**

时长 09:05 大小 8.33M



你好，我叫黄涛，目前在一家互联网公司从事大数据开发工作，主要负责大数据应用的开发，以及数仓平台工具的搭建和维护。

在跟随这门课程学习的过程中，我增长了很多大数据相关的知识，对于大数据技术和相关开源组件，也有了更深的了解。今天正好可以借着这个机会，来记录下自己的一点心得体会，也跟你分享一下我的学习思路，咱们一起聊一聊。

领资料

为什么我要来学这门课？



大数据在最近十几年非常红火，大数据技术及组件层出不穷。对于新入门大数据行业的人来说，会有一种进入少林藏经阁，却不知从哪里学起的感觉。对于已经进入行业两三年的人来说，也会有一种技术不断更新迭代，何时是个尽头的苦恼。

之前我在极客时间参加了《大数据训练营》的学习，课程结束，老师介绍了如何在大数据行业持续精进的方法，其中提到了一个关键，就是**去读大数据技术背后的论文**。

了解大数据的人都清楚，大数据技术的兴起，绕不开 Google 在十几年前发表的三篇论文。而第一代大数据技术，我们如雷贯耳的 HDFS、HBase、MapReduce，也正是基于这三篇论文实现的。大数据技术的迭代，背后也正是知名高校或大公司发表的相关论文。没有这些论文，这些技术也无从谈起。

当然，我们也都知道，论文可以说是知识浓缩后的精华。随便一段论文内容，也包含着大量的信息，想要读懂，极为不易。幸运的是，极客时间开设了一门课程，就是徐文浩老师的《大数据经典论文解读》课。我发现后，也是第一时间下单购买了课程，希望能在老师的带领下，深入理解大数据技术背后的这些论文。

学习课程的一些心得

老师在课程的 **第 2 讲** “学习方法：建立你的大数据知识网络” 中讲到，大数据技术从大的方面来讲，主要分为分布式系统、存储引擎、计算引擎。大数据技术，本身就是与传统数据处理技术不一样，最大的区别根源就在于这个“大”字上。

任何一个大数据系统，本质也是一个分布式系统。因此，我主要是从分布式系统的几个特点，来关注一篇论文的技术点。另外，任何一个系统，自然也有它本身突出的功能点。所以，我主要就是根据以下四点来学习一篇论文的：

- 可靠性（高可用）；
- 可扩展性（可伸缩）；
- 可维护性（容错性）；
- 功能。

这里呢，我也拿 GFS 和 Kafka 来做个示例。

GFS

就拿老师讲述 GFS 的那三讲来说，三节课都是在分析这篇论文《The Google File System》。GFS 在架构上，分别提供了目录服务的 Master，以及存储服务的

chunkserver，它们类似 HDFS 的 NameNode 和 DataNode。

数据在 chunkserver 的存储默认是三个副本，天然具备高可用的。而整个系统要提供高可用的服务，就得确保 Master 的高可用，于是 GFS 就通过 Shadow Master 来实现。当 Master 故障时，Shadow Master 会利用异步复制过来的数据，提供服务。

在可扩展性方面，因为 GFS 是存储引擎，主要提供存储服务，所以增加 chunkserver，也就增加了扩展性。GFS 是通过负载均衡技术，来使不同存储节点的磁盘利用率趋于平均水平。

另外在容错方面，GFS 是通过重放 Checkpoint 之后的操作日志，实现了 Master 重启后可快速恢复的目标。当然，Backup Master 也保障了即使硬件故障，也无需担心。

除了上述三点，GFS 也有一些功能，值得我们了解学习。比如，单 Master 设计简单，不用考虑选举问题；数据写入流程确保了数据一致性；流水线式网络数据传输，避免了网络瓶颈；控制流和数据流分离；Snapshot 复制指令，优化了文件复制性能。

Kafka

Kafka 是大数据领域很知名的一个组件，老师也花了两讲的时间带我们进行了分析。Kafka 的高可用，就在于每一个 Topic 分区，存在多个数据副本。即使当前的 Leader 出现了故障，我们也可以快速从 ISR 列表中选择一个新的分区作为 Leader，来对外提供服务。

而在扩展性方面，Kafka 的集群可动态调整 Broker 的数量。在容错方面，Kafka 的多副本分区，加上合适的配置，就很好地保障了当部分节点故障时，数据的一致性。

此外，Kafka 还有这些特性：单 Partition 读写；通过日志文件持久化数据，并使用索引文件和数据文件分开存储；对于消费者的负载均衡；保证消息在 Partition 的顺序；作为 Kappa 架构的重要组件，实现流批一体。

最后的一点感悟

其实，通过学习这门课，我也有几点感触，想要分享给你，如果也能跟你产生一些共鸣或者可以带来一些启发，就太好了。

好的系统就是在资源有限的情况下想出的优秀解决方案。

这些经典论文有的发表于十多年前，有的是几年前。虽然技术不断更新迭代，软硬件都在飞速发展，但是在任何时候，资源都是有限的。在这种情况下，更能体现出好架构的价值。这些经典论文和知名开源系统，正是呈现给我们很多这样的解决方案。

充分考虑系统的各种依赖条件。

作为软件工程师或者架构师，我们也需要考虑到各种依赖。在《数据密集型应用系统设计》一书讲到，Bug 都是源于我们软件的依赖条件未能得到满足。我们通常假设这些依赖条件成立，但是，任何事情都不是绝对的，特别是硬件资源。硬件的故障、资源的竞争等等，这些都可能导致 Bug 产生。因此，好的架构设计需要充分考虑各种依赖条件，并做好依赖条件失效时的应对之策，只有这样，才能保证系统的稳健运行。

不只考虑系统内部，还要考虑数据流的上下游。

任何大数据组件都不是孤立存在的，而是更大系统的一部分。因此，在设计系统时，我们要同时考虑到数据流的上下游，这样才能更好地融入大的生态，或者建立起自己的生态。

任何系统都是多种要素的均衡。

任何一个系统都不可能做到面面俱到，能做到有几个亮点，没有大的短板就足够了。大数据技术众多，类似的技术和组件也不少，我们通常都是根据自己的使用场景，结合各种技术的优缺点来进行选择。而我们很难能保证，同样出现的几个技术或组件，一个就能做到力压其他所有。我们在设计自己的系统时，实际上也是要有有所兼顾、有所放弃的。

大数据技术种类繁杂，涉及的知识多如牛毛，我们只有拿出水滴石穿的劲头，一步一个脚印，持续学习，稳扎稳打，不懈努力。同时，在学习一类新的技术时，仔细琢磨，直探本质，才能做到触类旁通，事半功倍。

在此，我也希望能跟一起学习这门课程的你一起努力，咱们多多交流，共同进步！

[生成海报并分享](#)[赞 1](#) [提建议](#)

© 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。

上一篇 用户故事 | 陈煌：唯有自强不息，方能屹立不倒

下一篇 用户故事 | 许灵：不抛弃不放弃

更多学习推荐

190 道大数据高频面试真题

涵盖 11 个核心技术栈 + 4 套大厂真题

免费领取 



精选留言

[写留言](#)

由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。