



下载APP

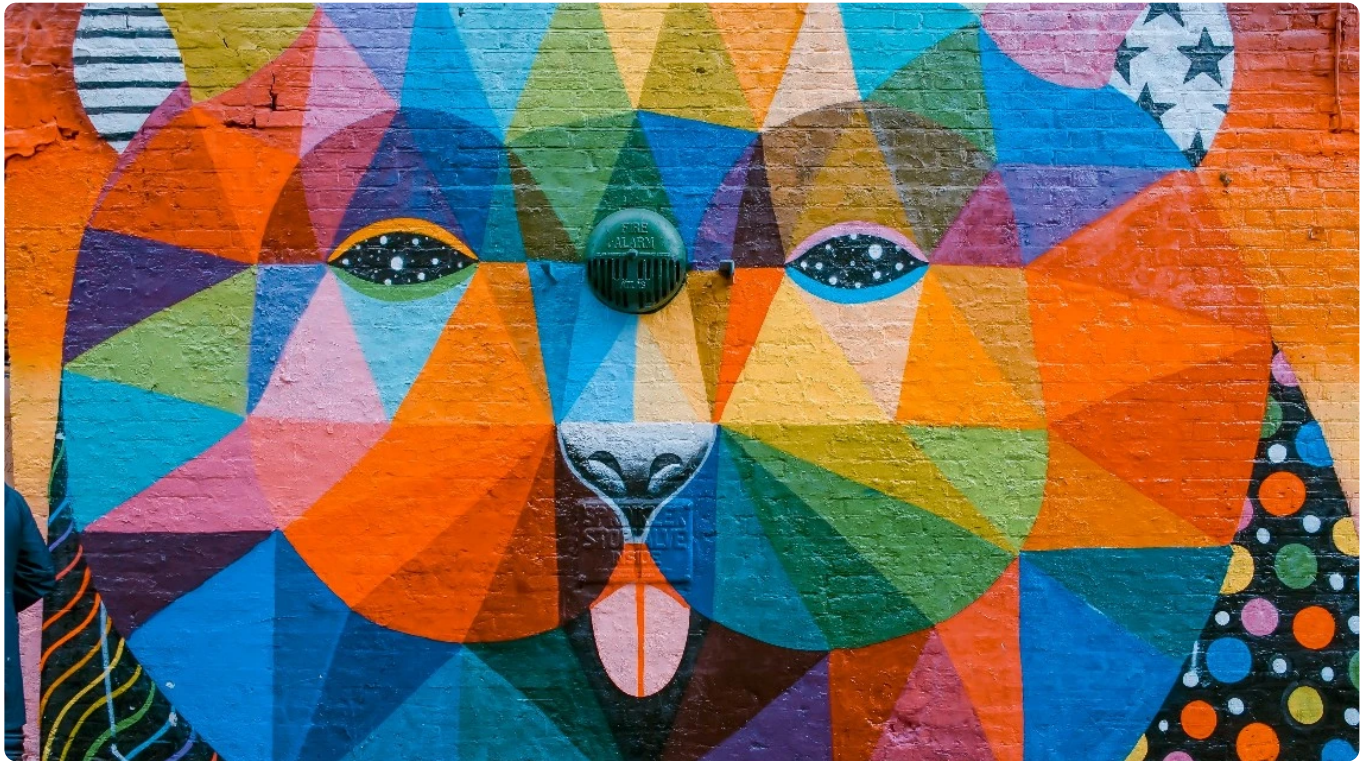


用户故事 | 核桃：95后的技术成长之路

2022-01-21 核桃

《大数据经典论文解读》

课程介绍 >



讲述：王惠

时长 09:59 大小 9.15M



你好，我是核桃，一名 95 后分布式文件系统工程师，目前在武汉工作。

2018 年我在大学毕业之前，就学习过大数据开发相关的课程内容，所以毕业的时候，理所当然就选择了做大数据开发相关的工作。在 2020 年的时候，我转为了分布式文件系统运维，负责 GlusterFS 和 ZFS 相关的系统维护。到了 2021 年，我转型成为分布式文件系统研发工程师，目前是负责自研底层文件系统模块功能研发。

领资料

为什么我要读大数据经典论文？



其实说来也有趣，大数据是我相对比较熟悉的专业领域，但是呢，也是我产生较多疑惑的领域。

还记得在刚毕业的时候，那时候 Hadoop 和 Spark 还是核心工具，然而到了今天，已经有不少大数据工程师喊出了“MapReduce 已凉”的口号，而对于 Spark，也一直有说被 Flink 取代的趋势。众说纷纭，但不管未来发展如何，至少我们可以看到，**在短短的几年内，整个大数据生态圈的发展，已经到了一个非常高速的阶段了。**

当然除了这些争论以外，现在大数据生态圈随着 SQL 化的出现，NoteBook+Spark 开发模式的成熟，还有一些图形化的机器学习平台出现，就会有个别工程师甚至悲观地认为，未来似乎对于大数据工具的发展，已经到了顶点。大数据现在已经“夕日无限好，只是近黄昏”，因为可以研究的方向基本上都稳定和固定了，未来除了极少数的底层研究，似乎并没有太多的发展方向了。

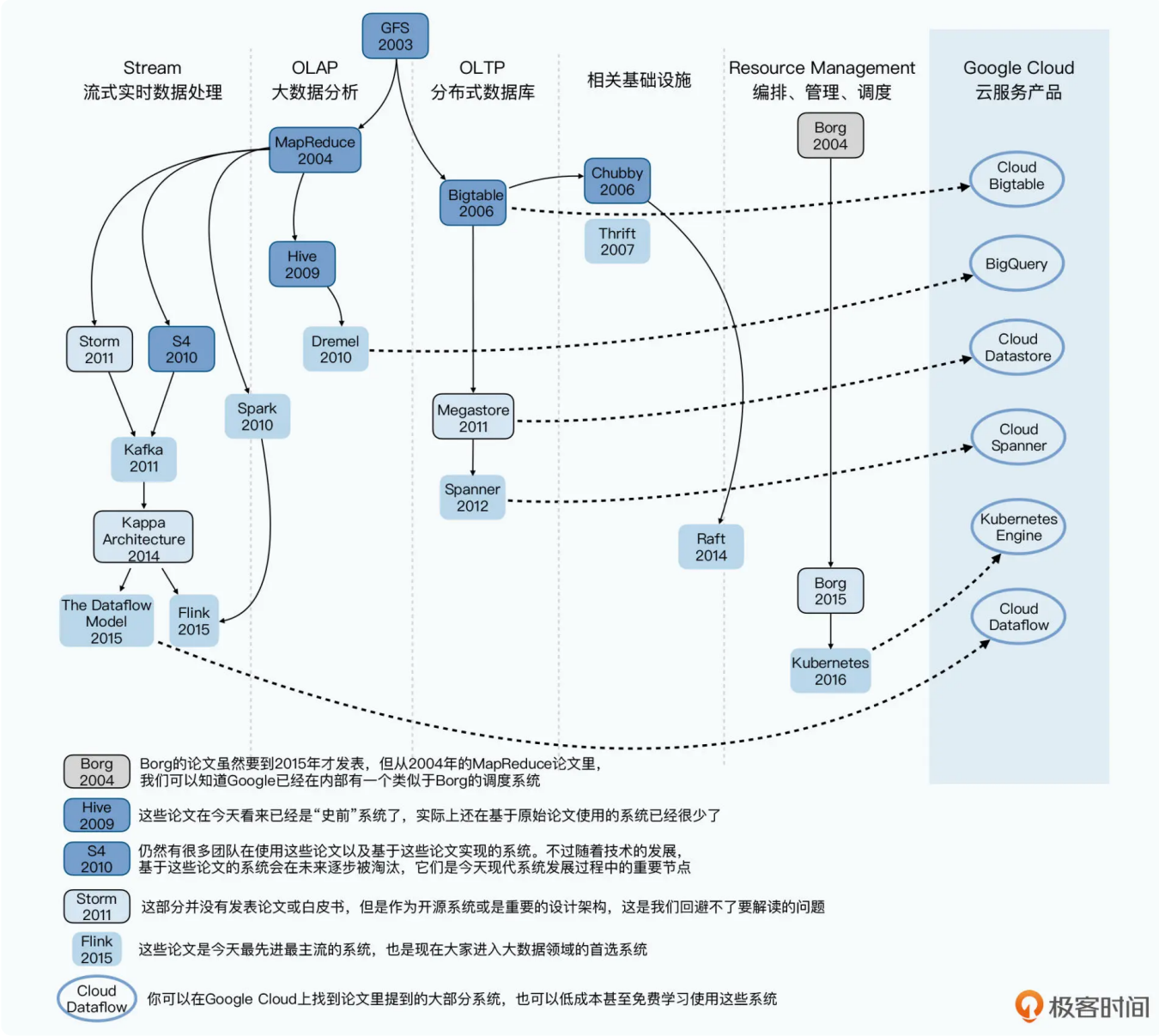
但是未来的发展真的是这样吗？我着实是困惑了很长一段时间。

带着这些困惑，以及个人对大数据技术理解的渴望，我一直在尝试着用不同的方式去明辨一些争论。比如我曾经尝试过读 Spark 相关源码的书籍（当时 2.4 还是主流的版本），但是限于个人基础的不扎实，光是理解一个 SparkContext 的生成启动过程，就花了不少时间。

同时，我那时候对于 RPC、Spark Streaming 的理解，也一直没有太深刻，所谓的批处理和流处理模式的不同，网上看到的资料，一直让我感觉似懂非懂。不是网上的资料不好，而是总感觉网上资料提到的内容，对我来说似乎是**隔了一层雾，需要找方法和方式来点破。**

后来我遇到了徐老师的《大数据经典论文的解读》课程，才明白，其实我点破这层迷雾的方式，就是**要去深入读一读这些经典论文**。而老师的课程，恰好有了一个平台和方式，能够让我在深入阅读这些论文之前，宏观地理解论文的核心内容。

在学习了徐老师的课程之后，我最喜欢的是 **第 1 讲** 关于 12 年大数据生态演变的内容，就是下面的这张架构图，这一系列的生态演变在徐老师的梳理之下，非常清晰。



另外还有一个有意思的点我想再和你分享下，在 2022 年元旦假期期间，我看了自己在极客时间的年度报告，在过去 2021 年我学习的课程数达到了 14 门，超过了 93% 的用户。我个人在极客时间订阅的课程数量是超过 20 门的，从 Linux 到计算机组成原理，从数据库实战再到 Rust 语言等，不同方向都有所涉猎，但是唯独关于经典论文这部分，目前似乎只有徐老师这一门课程，可以说是一枝独秀。

如何更好地读论文？

我后来因为换了方向，在过去的 2021 年下半年中，重新学习了分布式文件系统的内容，读了一些文件系统相关的论文，其中最经典之一的，莫过于 ZFS 最早的一篇设计论文，那是由 Jeff 写的，还有一篇是关于 Btrfs 的，讲解 Copy-on-write 和 Defragmentation 技术实现的论文，非常值得一读。

当然，现在关于大数据的经典论文，在网上也有了一部分翻译，但我还是非常建议你有机会有话，要亲自去读读原文，那样会有更加深刻的理解。同时我也想分享一下，个人关于读论文的建议和想法。当然，因为本人能力有限，部分观点如有不当之处，敬请谅解。

抛开对现在技术的执念

有不少经典论文因为时间太过于久远了（不少论文甚至是上世纪八九十年代的），所以我们在读到这些论文时，总是会不自觉地和现在版本技术做一个纵向比较。也可能读着读着会觉得，那时候的论文写的东西太简陋了，细节太少，或多或少带有各种主观想法在里面。你一旦这样想，就很容易有抵触心态了。

我们其实回过头看看在论文产生的年代，那个时候计算机发展没有现在的规模和场景，很多优化的方法和内容，都是在无数次技术版本迭代和尝试之后才确定的。所以，如果我们只是简单地以今天的要求来看论文，那么之前的这些经典论文统统都是不合格的。

但是经典论文为什么经典？一定是有**启发你的洞见和视角**在其中的。抛开对现在技术的执念，学会理解不同论文写就时的当下时代技术背景，这样才能真正融入到论文作者考虑的场景和思考问题的方式中。

模块化读论文

很多时候，论文在讲解某个技术时，都是讲了不同的模块的，语言非常精练简洁，同时讲解的内容很宏观，并且会有很多不同模块的实现交叉。但是这些模块缠绕在一起的时候，虽然对理解这个技术有很大的帮助，而我们却无法跳出这个工具，这样有时候就会有一种，一个技术工具一篇论文一种实现的想法，也就是**很难把某个模块技术单独抽象出来，进行横向对比**。

这样时间久了，其实会很痛苦。比如对存储类工具来说，从 HDFS 到 HBase 再到 MongoDB，虽然这是不同的工具，但是本质上不管是分布式文件系统还是分布式数据库，都是用来存数据的。那么一致性的内容应该都是具备的，功能也是相似的。

而有了一致性之后，在底层存储时，数据的安全性和稳定性实现，比如如何保证事务或者一些读写请求的可靠执行，思想上是具备相似之处的，不管是通过各种日志、还是一些文件扩展属性来标记（GlusterFS 为代表），本质上的作用和功能都是类似的。

在这样的思想指导下，目前也有了一种数据库与存储不分家的说法。当然，在学习徐老师的课程时，其实我们也可以很明显地感受到这种**分模块化阅读**的感觉。所以我建议你读完徐老师课程中，关于 GFS 和 Bigtable 的内容之后，来横向对比一下，然后再结合 HDFS 的实现，去看看在一致性、底层存储方面的差异和区别，这样印象就会非常深刻了。

论文与工程实现的结合

读了论文和很多技术资料之后，终究还是要应用到现实的工程实践中的。而在这两者的结合上，这个时候，我们可能就需要阅读大量的各种版本功能实现的一些代码思想和资料了。这里我就拿我阅读 OpenZFS 的源码为例。

在我读了论文后，去阅读最新的代码时，我会首先选择一个模块的数据结构来阅读，比如 ZFS 管理磁盘空间的一个模块叫 Metaslab（和 Linux 的 Slab 有点不同，虽然当年 Jeff 也是该设计者）。

然后，找到这个模块代码的数据结构，去了解在 Metaslab 中，Allocator 和 Object 之间的关系等等。同时，因为国外很多优秀的开源项目，大部分对于代码注释和场景注释非常多，这时候我会通过数据结构入手，这样就可以更加形象化和具体化地理解磁盘管理的实现。通过这样的方式，我再一步步地去查看不同数据结构的函数调用和改变，就可以把模块流程串联起来了。

总结

前面说了很多内容，其实回到最开始提到的，在学习了徐老师的课程之后，我也更好地理解了一些技术的发展背景，觉得自己也算是“拨云见日”了。

我个人目前对大数据的未来发展，是有非常高的期待和热情的。随着基于内存的分布式系统 Alluxio、云原生的 OpenEBS 出现，存储接口大一统的发展，还有大数据工具结合 Kubernetes 的工具，又或者打通与 TensorFlow 的壁垒等，在未来，大数据领域会有更加急速的发展与演变。而在这个时代到来之前，需要我们不断地对过去的技术溯本清源，这样才能更好地一步步脚踏实地发展和前进。

对了，最后分享一下我在 2021 年写的一本开源的 [📖 技术书](#)吧，关于 GlusterFS 系统的相对全面的学习内容，希望你会喜欢，谢谢~

不知道你学习这个课程的过程是怎样的呢？有没有什么独特的学习方法和心路历程呢？欢迎你写在留言区，我们一起分享，相互鼓励，共同进步！

分享给需要的人，Ta订阅本课程，你将得 20 元

生成海报并分享

赞 0 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 | 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。