82 | 程序员练级攻略：分布式架构经典图书和论文

2018-07-12 陈皓

左耳听风 进入课程 >

讲述：柴巍
时长 17:17 大小 7.92M

经典图书

首先，我推荐几本分布式架构方面的经典图书。

Distributed Systems for fun and profit，这是一本免费的电子书。作者撰写此书的目的是希望以一种更易于理解的方式，讲述以亚马逊的 Dynamo、谷歌的 Bigtable 和 MapReduce 等为代表的分布式系统背后的核心思想。

Designing Data Intensive Applications，这本书是一本非常好的书，我们知道，在分布式的世界里，数据结点的扩展是一件非常麻烦的事。这本书深入浅出地用很多的工程案例讲解了如何让数据结点做扩展。作者马丁·科勒普曼（Martin Kleppmann）在分布式数据系统领域有着很深的功底，并在这本书中完整地梳理各类纷繁复杂设计背后的技术逻辑，不同架构之间的妥协与超越，很值得开发人员与架构设计者阅读。

这本书深入到 B-Tree、SSTables、LSM 这类数据存储结构中，并且从外部的视角来审视这些数据结构对 NoSQL 和关系型数据库的影响。这本书可以让你很清楚地了解到真正世界的大数据架构中的数据分区、数据复制的一些坑，并提供了很好的解决方案。最赞的是，作者将各种各样技术的本质非常好地关联在一起，令你触类旁通。
而且，这本书完全就是抽丝剥茧，循循善诱，从“提出问题”到“解决问题”、“解决方案”、“优化方案”和“对比不同的方案”，一点一点地把非常晦涩的技术和知识展开。本书的引用相当多，每章后面都有几百个 Reference，通过这些 Reference 你可以看到更为广阔、更为精彩的世界。

Distributed Systems: Principles and Paradigms ，本书是由计算机科学家安德鲁·斯图尔特·塔能鲍姆（Andrew S. Tanenbaum）和其同事马丁·范·斯蒂恩（Martin van Steen）合力撰写的，是分布式系统方面的经典教材。
语言简洁，内容通俗易懂，介绍了分布式系统的七大核心原理，并给出了大量的例子；系统讲述了分布式系统的概念和技术，包括通信、进程、命名、同步化、一致性和复制、容错以及安全等；讨论了分布式应用的开发方法（即范型）。但本书不是一本指导“如何做”的手册，仅适合系统性地学习基础知识，了解编写分布式系统的基本原则和逻辑。中文翻译版为《分布式系统原理与范型》（第二版）。

Scalable Web Architecture and Distributed Systems，
这是一本免费的在线小册子，其中文翻译版 可扩展的 Web 架构和分布式系统。本书主要针对面向互联网（公网）的分布式系统，但其中的原理或许也可以应用于其他分布式系统的设计中。作者的观点是，通过了解大型网站的分布式架构原理，小型网站的构建也能从中受益。本书从大型互联网系统的常见特性，如高可用、高性能、高可靠、易管理等出发，引出了一个类似于 Flickr 的典型的大型图片网站的例子。

Principles of Distributed Systems ，本书是苏黎世联邦理工学院的教材。它讲述了多种分布式系统中会用到的算法。虽然分布式系统的不同场景会用到不同算法，但并不表示这些算法都会被用到。不过，作为学生来说，掌握了算法设计的精髓也就能举一反三地设计出解决其他问题的算法，从而得到分布式系统架构设计中所需的算法。

经典论文

分布式事务

想了解分布式模型中最难的“分布式事务”，你需要看看 Google App Engine 联合创始人瑞恩·巴雷特（Ryan Barrett）在 2009 年的 Google I/O 大会上的演讲《Transaction Across DataCenter》（YouTube 视频）。

在这个演讲中，巴雷特讲述了各种经典的解决方案如何在一致性、事务、性能和错误上做平衡。而最后得到为什么分布式系统的事务只有 Paxos 算法是最好的。

下面这个图是这个算法中的结论。

你也可以移步看一下我在 Coolshell 上写的这篇文章《分布式系统的事务处理》。

Paxos 一致性算法

Paxos 算法，是莱斯利·兰伯特（Lesile Lamport）于 1990 年提出来的一种基于消息传递且具有高度容错特性的一致性算法。但是这个算法太过于晦涩，所以一直以来都属于理论上的论文性质的东西。其真正进入工程圈，主要是来源于 Google 的 Chubby lock——一个分布式的锁服务，用在了 Bigtable 中。直到 Google 发布了下面这两篇论文，Paxos 才进入到工程界的视野中来。

> Bigtable: A Distributed Storage System for Structured Data
>
> The Chubby lock service for loosely-coupled distributed systems

Google 与 Bigtable 相齐名的还有另外两篇论文。

> The Google File System
>
> MapReduce: Simplified Data Processing on Large Clusters

不过，这几篇文章中并没有讲太多的 Paxos 算法上的细节，反而是在Paxos Made Live - An Engineering Perspective 这篇论文中提到了很多工程实现的细节。这篇论文详细解释

了 Google 实现 Paxos 时遇到的各种问题和解决方案，讲述了从理论到实际应用二者之间巨大的鸿沟。

Paxos 算法的原版论文比较晦涩，也不易懂。这里推荐一篇比较容易读的—— Neat Algorithms - Paxos 。这篇文章中还有一些小动画帮助你读懂。还有一篇可以帮你理解的文章是 Paxos by Examples。

Raft 一致性算法

因为 Paxos 算法太过于晦涩，而且在实际的实现上有太多的坑，并不太容易写对。所以，有人搞出了另外一个一致性的算法，叫 Raft。其原始论文是 In search of an Understandable Consensus Algorithm (Extended Version) ，寻找一种易于理解的 Raft 算法。这篇论文的译文在 InfoQ 上，题为《Raft 一致性算法论文译文》，推荐你读一读。

这里推荐几个不错的 Raft 算法的动画演示。

Raft - The Secret Lives of Data

Raft Consensus Algorithm

Raft Distributed Consensus Algorithm Visualization

Gossip 一致性算法

后面，业内又搞出来一些工程上的东西，比如 Amazon 的 DynamoDB，其论文Dynamo: Amazon’ s Highly Available Key Value Store 的影响力非常大。这篇论文中讲述了 Amazon 的 DynamoDB 是如何满足系统的高可用、高扩展和高可靠的。其中展示了系统架构是如何做到数据分布以及数据一致性的。GFS 采用的是查表式的数据分布，而 DynamoDB 采用的是计算式的，也是一个改进版的通过虚拟结点减少增加结点带来数据迁移的一致性哈希。

这篇文章中有几个关键的概念，一个是 Vector Clock，另一个是 Gossip 协议。

Time, Clocks and the Ordering of Events in a Distributed System ，这篇文章是莱斯利·兰伯特（Leslie Lamport）于 1978 年发表的，并在 2007 年被选入 SOSP 的名人堂，被誉为第一篇真正的“分布式系统”论文，该论文曾一度成为计算机科学史上被引用最多的文章。分布式系统中的时钟同步是一个非常难的问题，因为分布式系统中是使用消

息进行通信的，若使用物理时钟来进行同步，一方面是不同的 process 的时钟有差异，另一方面是时间的计算也有一定的误差，这样若有两个时间相同的事件，则无法区分它们谁前谁后了。这篇文章主要解决分布式系统中的时钟同步问题。

马萨诸塞大学课程 Distributed Operating System 中第 10 节 Clock Synchronization，这篇讲义讲述了时钟同步的问题。

关于 Vector Clock，你可以看一下 Why Vector Clocks are Easy 和 Why Vector Clocks are Hard 这两篇文章。

用来做数据同步的 Gossip 协议的原始论文是 Efficient Reconciliation and Flow Control for Anti-Entropy Protocols。Gossip 算法也是 Cassandra 使用的数据复制协议。这个协议就像八卦和谣言传播一样，可以“一传十、十传百”传播开来。但是这个协议看似简单，细节上却非常麻烦。

Gossip 协议也是 NoSQL 数据库 Cassandra 中使用到的数据协议，你可以上 YouTube 上看一下这个视频介绍： Understanding Gossip (Cassandra Internals)。

关于 Gossip 的一些图示化的东西，你可以看一下动画 Gossip Visualization。

分布式存储和数据库

除了前面的 Google 的 BigTable 和 Google File System 那两篇论文，还有 Amazon 的 DynamoDB 的论文，下面也有几篇也是要读一下的。

一篇是 AWS Aurora 的论文 Amazon Aurora: Design Considerations for High Throughput Cloud -Native Relation Databases。

另一篇是比较有代表的论文是 Google 的 Spanner: Google’ s Globally-Distributed Database。 其 2017 年的新版论文：Spanner, TrueTime & The CAP Theorem。

F1 - The Fault-Tolerant Distributed RDBMS Supporting Google’ s Ad Business 。

Cassandra: A Decentralized Structured Storage System 。

CRUSH: Controlled, Scalable, Decentralized Placement of Replicated Data, 这里提到的算法被应用在了 Ceph 分布式文件系统中，其架构可以读一下 RADOS - A Scalable, Reliable Storage Service for Petabyte-scale Storage Clusters 以及 Ceph 的架构文档。

分布式消息系统

分布式消息系统，你一定要读一下 Kafka 的这篇论文 Kafka: a Distributed Messaging System for Log Processing。

Wormhole: Reliable Pub-Sub to Support Geo-replicated Internet Services ，Wormhole 是 Facebook 内部使用的一个 Pub-Sub 系统，目前还没有开源。它和 Kafka 之类的消息中间件很类似。但是它又不像其它的 Pub-Sub 系统，Wormhole 没有自己的存储来保存消息，它也不需要数据源在原有的更新路径上去插入一个操作来发送消息，是非侵入式的。其直接部署在数据源的机器上并直接扫描数据源的 transaction logs，这样还带来一个好处，Wormhole 本身不需要做任何地域复制（geo-replication）策略，只需要依赖于数据源的 geo-replication 策略即可。

All Aboard the Databus! LinkedIn’s Scalable Consistent Change Data Capture Platform ，在 LinkedIn 投稿 SOCC 2012 的这篇论文中，指出支持对不同数据源的抽取，允许不同数据源抽取器的开发和接入，只需该抽取器遵循设计规范即可。该规范的一个重要方面就是每个数据变化都必须被一个单调递增的数字标注（SCN），用于同步。这其中的一些方法完全可以用做异地双活的系统架构中。（和这篇论文相关的几个链接如下：PDF 论文 、 PPT 分享。）

日志和数据

The Log: What every software engineer should know about real-time data’s unifying abstraction ，这篇文章好长，不过这是一篇非常好非常好的文章，这是每个工程师都应用知道的事，必看啊。你可以看中译版《日志：每个软件工程师都应该知道的有关实时数据的统一概念》。

The Log-Structured Merge-Tree (LSM-Tree) ，N 多年前，谷歌发表了 Bigtable 的论文，论文中很多很酷的方面，其一就是它所使用的文件组织方式，这个方法更一般的名字叫 Log Structured-Merge Tree。LSM 是当前被用在许多产品的文件结构策略：HBase、Cassandra、LevelDB、SQLite，甚至在 MongoDB 3.0 中也带了一个可选的 LSM 引擎（Wired Tiger 实现的）。LSM 有趣的地方是它抛弃了大多数数据库所使用的传统文件组织方法。实际上，当你第一次看它时是违反直觉的。这篇论文可以让你明白这个技术。（如果读起来有些费解的话，你可以看看中文社区里的这几篇文章：文章一、文章二。）

Immutability Changes Everything ，这篇论文是现任 Salesforce 软件架构师帕特·赫兰德（Pat Helland）在 CIDR 2015 大会上发表的（相关视频演讲）。

Tango: Distributed Data Structures over a Shared Log）。这个论文非常经典，其中说明了不可变性（immutability）架构设计的优点。随着为海量数据集存储和计算而设计的以数据为中心的新型抽象技术的出现，分布式系统比以往任何时候都更容易构建。但是，对于元数据的存储和访问不存在类似的抽象。
为了填补这一空白，Tango 为开发人员提供了一个由共享日志支持的内存复制数据结构（例如地图或树）的抽象。Tango 对象易于构建和使用，通过共享日志上简单的追加和读取操作来复制状态，而不是复杂的分布式协议。在这个过程中，它们从共享日志中获得诸如线性化、持久性和高可用性等属性。Tango 还利用共享日志支持跨不同对象的快速事务处理，允许应用程序跨机器进行状态划分，并在不牺牲一致性的情况下扩展到底层日志的上限。

分布式监控和跟踪

Google 的分布式跟踪监控论文 - Dapper, a Large-Scale Distributed Systems Tracing Infrastructure， 其开源实现有三个 Zipkin、Pinpoint 和 HTrace。我个人更喜欢 Zipkin。

数据分析

The Unified Logging Infrastructure for Data Analytics at Twitter ，Twitter 公司的一篇关于日志架构和数据分析的论文。

Scaling Big Data Mining Infrastructure: The Twitter Experience ，讲 Twitter 公司的数据分析平台在数据量越来越大，架构越来越复杂，业务需求越来越多的情况下，数据分析从头到底是怎么做的。

Dremel: Interactive Analysis of Web-Scale Datasets，Google 公司的 Dremel，是一个针对临时查询提供服务的系统，它处理的是只读的多层数据。本篇文章介绍了它的架构与实现，以及它与 MapReduce 是如何互补的。

Resident Distributed Datasets: a Fault-Tolerant Abstraction for In-Memory Cluster Computing，这篇论文提出了弹性分布式数据集（Resilient Distributed Dataset，RDD）的概念，它是一个分布式存储抽象，使得程序员可以在大型集群上以容错的方式执行内存计算；解释了其出现原因：解决之前计算框架在迭代算法和交互式数据挖掘工具两种应用场景下处理效率低下的问题，并指出将数据保存在内存中，可以将性能提高一个数量级；同时阐述了其实现原理及应用场景等多方面内容。很有趣儿，推荐阅读。

与编程相关的论文

Distributed Programming Model

PSync: a partially synchronous language for fault-tolerant distributed algorithms

Programming Models for Distributed Computing

Logic and Lattices for Distributed Programming

其它的分布式论文阅读列表

除了上面上的那些我觉得不错的论文，下面还有三个我觉得不错的分布式系统论文的阅读列表，你可以浏览一下。

Services Engineering Reading List

Readings in Distributed Systems

Google Research - Distributed Systems and Parallel Computing

小结

今天分享的内容是分布式架构方面的经典图书和论文，并给出了导读文字，几乎涵盖了分布式系统架构方面的所有关键的理论知识。这些内容非常重要，是学好分布式架构的基石，请一定要认真学习。

下篇文章中，我们将讲述分布式架构工程设计方面的内容，包括设计原则、设计模式以及工程实践等方面的内容。敬请期待。

下面是《程序员练级攻略》系列文章的目录。

专业基础篇

- 编程语言
- 理论学科
- 系统知识

软件设计篇

- 软件设计

高手成长篇

- Linux 系统、内存和网络（系统底层知识）
- 异步 I/O 模型和 Lock-Free 编程（系统底层知识）
- Java 底层知识
- 数据库
- 分布式架构入门（分布式架构）
- 分布式架构经典图书和论文（分布式架构）
- 分布式架构工程设计 (分布式架构)
- 微服务
- 容器化和自动化运维
- 机器学习和人工智能
- 前端基础和底层原理（前端方向）
- 前端性能优化和框架（前端方向）
- UI/UX 设计（前端方向）
- 技术资源集散地

上一篇　81 | 程序员练级攻略：分布式架构入门

下一篇　83 | 程序员练级攻略：分布式架构工程设计

精选留言 (13)

写留言

虎虎
2018-07-12

27

这些论文和书读了有一半了，但是还是进不了分布式的门。因为全部是自己兴趣学的，没有相关开发经验。感觉找个分布式开发的工作挺难的。望指点迷津。

展开

A
2018-07-16

5

太高级了。前面几篇就够我实践几年了，这些完全看不来了，要不我们再谈谈人生理想。。

慕容引刀
2018-07-12

2

果然对于编程最难的还是英文😂

展开

周扬
2018-07-14

1

建议皓叔讲讲自己公司的开发，自动化测试，自动化部署，自动化运维的流程。

墨梵
2018-07-13

1

靠自己兴趣学的分布式，假设读完上述的文章和书籍，在有了理论指导前提下，如何获取实际相关开发经验（最终目的是找个分布式开发的工作，前提是无分布式相关开发经验的小白）求耗叔指点...

godtrue
2019-01-10

太棒了，够消化许多年了，幸好还年轻有时间和兴趣，英文要么是门槛，要么阶梯。

小蛋壳
2018-07-14

在实际研发中，梳理需求，模型设计等是日常工作的常态，怎么提高这一块？

破晓
2018-07-13

都是英文的，读起来有点吃力和缓慢。

展开

ydp
2018-07-13

耗子老师，希望后面在讲讲，大数据和网络安全等内容。谢谢

刘宝峰_DEV
2018-07-13

希望陈老师能出一篇关于单元测试，集成测试，和代码重构方面的文章😁

创意
2018-07-13

分布式我是在写程序中碰到名种问题，看了很多资料和书，很依然像画画一样，太多太多的未知

sevenfan
2018-07-13

非常全面，赞

展开

永光
2018-07-13

建议后续讲点机器学习、人工智能、深度学习相关知识，非常感谢。