

分布式系统原理

任杰

2020 年 5 月 12 日

目录

| | |
|---------------|----|
| 第一章 基础概念 | 7 |
| 1.1 认识分布式系统 | 7 |
| 1.2 什么是系统 | 7 |
| 1.3 为什么要做分布式 | 8 |
| 第二章 时间 | 9 |
| 2.1 物理时间 | 9 |
| 2.2 逻辑时间 | 9 |
| 第三章 一致性 | 11 |
| 3.1 顺序一致性 | 11 |
| 3.2 线性一致性 | 11 |
| 3.3 ACID | 11 |
| 3.4 最终一致性 | 11 |
| 第四章 共识算法 | 13 |
| 4.1 定义 | 13 |
| 4.2 Raft | 13 |
| 4.3 Paxos 及其它 | 13 |
| 4.4 复制状态机 | 13 |
| 第五章 数据复制 | 15 |
| 5.1 同步复制 | 15 |
| 5.2 异步复制 | 15 |
| 5.3 Quorum | 15 |
| 5.4 一致性复制 | 15 |
| 第六章 数据切割 | 17 |
| 6.1 切割原则 | 17 |
| 6.2 一致性哈希 | 17 |
| 第七章 混沌工程 | 19 |
| 第八章 分布式系统案例 | 21 |
| 8.1 TCP | 21 |
| 8.2 DNS | 21 |
| 8.3 CDN | 21 |
| 8.4 Spanner | 21 |

8.5 Kafka 21

序

十多年前我第一次走出校园正式从事软件开发工作。当时的一个任务是写一个批处理程序。这个程序有三个进程，需要在三台不同机器上运行。这些进程之间有先后依赖关系，需要按照特定的顺序启动。当时还没有 zookeeper，也没有 etcd，Raft 算法还没有被发明。当时试了很多方法都没有作用。最后只能靠人来手动维护进程状态。如果发现进程因为启动顺序错误报警的话手动重启正确的进程。

之后我就换了组，慢慢的就把这件事情遗忘了。大概一年多以后我突然想起了这个曾经让我头疼的问题。于是我请教了现在维护这个程序的开发人员。他很开心的给我展示了他的新解决方案。他做了一个很小的改动。之前每个进程如果发现所依赖的进程没有启动的话会出错退出。他把出错逻辑改成了一直重启，这样总有一次会有一个进程能启动正确，多来几次的话三个进程都能启动正确。我想了一想好像是这个道理。当时有一点不甘心。我可以写操作系统和编译器，却为什么连三个进程怎么正确启动都做不到。

多年以后我带着这个疑问进入了互联网的世界。这时候才发现我之前碰到的是分布式系统的一个简单到不能再简单的问题。分布式系统有它特定的领域问题和解决方案。对于做工程的人来说大多数情况下这些问题都有固定的解题套路。但是市面上缺乏能把问题解释的比较清楚的教材。这便是这本书背后的动机。

任杰

上海浦东

2020 年 5 月

第一章 基础概念

A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable

Leslie Lamport

Email message sent to a DEC SRC bulletin board at 12:23:29 PDT on 28 May 87.

1.1 认识分布式系统

一提到分布式系统大家第一个想到的就是互联网。在互联网上大家访问的每个主页背后都有对应的服务器。这些服务器通常都在某个数据中心或者电信机房。所有这些服务器一起提供了互联网服务。大家用到的手机也是电信通讯这个大的分布式系统的一部分。现在大家提到的物联网或者 IOT 也是一个大的分布式系统。这些熟知的分布式系统或多或少都和互联网有关系。那么有没有其它类型的分布式系统呢？

电力系统是一个隐藏很深的分布式系统。和之前提到的互联网、物联网、电信网不同，电力系统几乎是一个纯粹的物理世界的网络。电网有很多发电端，比如火电、风电、水电、太阳能等。这些电能会通过电网输送至千家万户。电网和互联网有很多相似之处。比如都有负载均衡和流量控制。当某些节点出现故障时系统能自行调度。其实电网和互联网的区别只是电子和电磁波的区别，两者甚至连传播的速度都是一样的。

分布式系统有一些共性，可以从分布式系统这个名词的组成看出。分布式系统这个名词有两个组成部分：分布和系统。分布式意味着存在多个功能单元分散在不同的地方。系统意味着所有这些功能单元能有机地结合在一起完成某个特定的任务。接下来让我们看看什么是系统和为什么要做分布式。

1.2 什么是系统

我们把一组单元称为一个系统并不仅仅因为它们仅仅有相似的名字或者颜色。系统和组织一样，都有既定的任务，比如互联网能让用户上网，电网能输电。这些任务通常都有质量要求。比如大家玩网络游戏会关心网络延时，多个人在一幢楼里下载文件时会关心下载的速度快不快。更专业一点的人会有量化的衡量指标，比如延时在百分之 99 的情况下是多少毫秒，即 P99 是多少。因此当我们在设一个分布式系统的时候，首先需要想清楚的是这个系统究竟是用来干什么的，其次是有没有可能用数字来衡量这个系统的质量。

1.3 为什么要做分布式

按照存在即合理¹的解释，一个系统之所以演变成为了分布式系统，那么它就一定有它背后原因。这个原因如此之强以至于就算分布式系统再难做也得做。一般来讲背后的原因有三个：

1. 物理学的资源稀缺性
2. 经济学的边际效益递减
3. 经济学的性价比权衡

拿一台电脑来举例。抽象来讲一个程序为了运行需要数据和计算资源。数据需要存储在计算机上，首选是存储在内存里。不同容量内存有不同价格。当我们需要单条内存 10 倍内存容量时，这个内存价格通常不是 10 倍而是更高。原因是单条内存容量越高，离物理局限越近，成品率越低，价格也越高，这就是物理资源稀缺性。这意味着额外的一块钱带来的内存容量提升会低于之前一块钱的效果，这就是边际效益递减。因此与其说我把这一块钱投资到已经很大的内存条上，还不如再买一块新的内存。主板在支持内存条数量这个问题上也有边际效应递减的问题，因此当主板支持一定数量的内存条后再增加额外的内存条就不合算了。这时候需要考虑把数据存储在硬盘上，因为硬盘更大更便宜。但如果硬盘的所有属性均优于内存，那么内存应该会慢慢退出消费市场无人购买。那么内存既然依然存在，硬盘必然有一些劣势，那就是硬盘速度大幅低于内存速度，这便是性价比权衡。我们选择了更大的容量，但是牺牲了数据处理速度。当数据量继续提升至硬盘的大小极限时，硬盘需要用磁带机来替换，但是这样数据处理速度会更慢。可选的单机存储方案是有限的，总存在一个数据量极限，超过这个极限之后速度慢的无法接受。这时候必须得考虑用多台机器来存储数据。

计算资源也存在同样的规律。单块 CPU 处理速度因为光速的原因存在极限。这时候需要考虑多块 CPU 来处理数据。但是主板能支持的 CPU 个数通常是有限的。和数据存储碰到的问题一样，当我们对 CPU 处理速度达到一定要求之后必须考虑用多台机器来处理。其它计算资源也面临同样的问题，比如网卡、GPU、路由器。

人们通常会低估单机处理能力，同时高估自己解决分布式系统的能力。因此当我们面临系统设计的问题时，能用单台机器解决就用单台机器，纵向扩容直至财务预算无法满足需求。接下来才能考虑横向扩展，用分布式系统解决方案。

¹“存在即合理”是对黑格尔的误解。黑格尔的原话是“Was vernünftig ist, das ist wirklich; und was wirklich ist, das ist vernünftig”，即“凡是合乎理性的东西都是现实的；凡是现实的东西都是合乎理性的”。

第二章 时间

2.1 物理时间

GPS, NTP

2.2 逻辑时间

Lamport Time, Vector clock

第三章 一致性

3.1 顺序一致性

3.2 线性一致性

3.3 ACID

3.4 最终一致性

第四章 共识算法

4.1 定义

4.2 Raft

4.3 Paxos 及其它

4.4 复制状态机

第五章 数据复制

5.1 同步复制

5.2 异步复制

5.3 Quorum

5.4 一致性复制

第六章 数据切割

6.1 切割原则

6.2 一致性哈希

第七章 混沌工程

第八章 分布式系统案例

8.1 TCP

8.2 DNS

8.3 CDN

8.4 Spanner

8.5 Kafka

参考文献

索引

分布式, 7

分布式系统, 7

系统, 7