

## 47 | 服务治理的宏观视角

2019-10-08 许式伟

许式伟的架构课

[进入课程 >](#)



讲述：许式伟

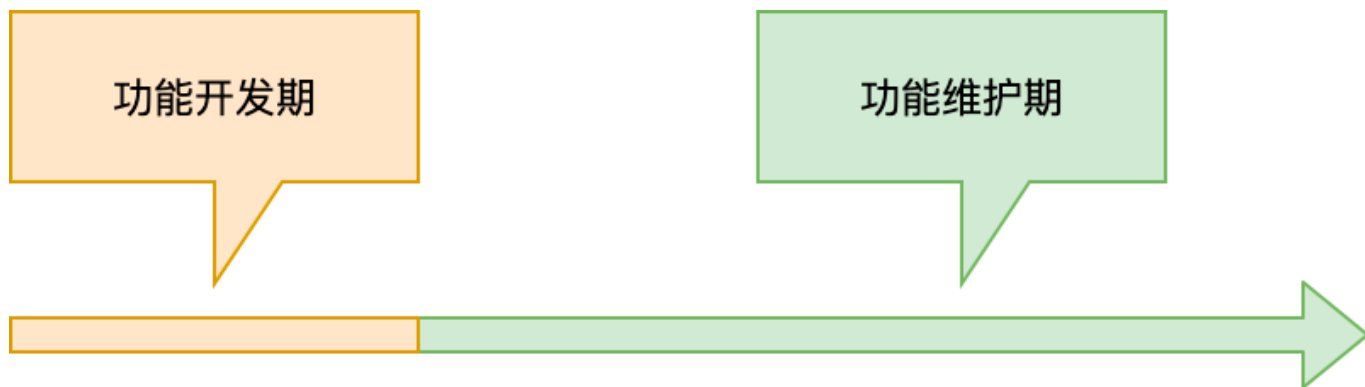
时长 08:37 大小 7.91M



你好，我是七牛云许式伟。

### 服务治理的目标

很多开发人员可能会习惯地认为，把软件开发出来交付给用户是其工作的结束。但实际上对于任何一个产品或者产品里面的某项功能来说，把东西开发出来只是个开始，实际上这个产品或功能在其被取代或去除之前，都会有很长一段时间的维护期。



上图是很基础的产品或功能的生命周期示意图。它并不只是对软件适用，而是对所有的商品适用。我们后面在“软件工程篇”中还会进一步探讨它。

对于这个示意图，我们核心需要理解的是两点：

其一，虽然功能开发阶段的成本是非常显性的，但是功能维护期，包括了功能迭代和售后维保，它的隐性成本往往更高。

其二，产品的功能开发期虽然有可能很短，但是它是起点，是源头。它每一分每一秒时间是怎么花的，很大程度上决定了这个产品或功能的最终维护代价。

互联网的诞生，对今天我们的生活产生了翻天覆地的影响。虽然细究起来它进入民用市场还只有短短二十多年的历史，但它的发展速度只能以“恐怖”来形容。

以互联网为载体的软件，它不只是在功能上要满足用户需求，还要提供健康的 24 小时不间断的服务。功能开发与维护的边界变得模糊，一些公司甚至每天都在发布新的版本。

要做到 24 小时不间断服务，这并不是那么容易的一件事情。

我们知道，传统意义上的操作系统，实现的是软件治理，它们的关注点是如何让众多的软件一起融洽相处，感觉上好像自己在独享着物理的硬件资源。

而服务治理的核心目标，除了软件治理外，更重要的是考虑如何确保这些软件能够真正做到 24 小时不间断的服务。

而这，才是服务端操作系统的使命。

## 服务治理系统

在上一讲，我们已经介绍了部分提供 24 小时不间断的服务所带来的挑战。但我们上一讲的侧重点在业务架构，所以我们主要关注点放在了对业务架构产生重要影响的内容，比如负载均衡和存储中间件。

从服务治理角度来说，把软件做出来只是一个开始。接下来我们面对的第一件事情，是如何把它发布出去。这就需要涉及部署、升级和版本管理等相关的话题。

软件在线上成功跑了起来，为用户提供了服务，我们接着面临的挑战是怎么保证它不会挂掉。这涉及非常多层面的事情。

首先是怎么知道服务是不是挂了，这就涉及监控与报警。在发现服务挂掉后，需要考虑尽快把它重启起来，恢复到正常的状态。

微观上某个软件进程挂掉不能影响到正常的服务。所以我们需要考虑各类故障域，尽可能全面地把单点故障的风险消除掉。

单点故障消除，有可能会是个运维问题，但更多时候我们也得从软件的业务架构层面去解决它。

服务治理并没有那么简单纯粹。虽然在理想情况下我们应该尽可能自动化所有故障的恢复，但故障的可能性太多，很多时候是我们无法提前预知的，这意味着人工介入无可避免。

所以，互联网不只是产生了服务端开发这样的工种，同时也产生了运维，或者说业务 SRE 这样的工种。

SRE 全称是 Site Reliability Engineer (网站可靠性工程师)，这是 Google 引入的一个职位，后被各类公司所借鉴。区别于传统意义上的运维，SRE 也是一个特殊的工程师群体，和服务端开发一样，他们肩负着自己独特的使命。

从服务端近年来的发展来看，产业进化的方向无不与服务治理相关：如何保证服务 24 小时不间断地运行。

故障基本上是难于避免的。可以导致故障的因素非常多。我们大体可以分为这么几个层面。

其一，软硬件升级与各类配置变更。变更是故障的第一大问题源头。保证系统不出问题的最简单的方法当然是不去升级。

但从用户的服务体验和竞争力的角度来说，升级又是必需的。所以这是一个服务端开发与 SRE 之间做平衡的问题。

其二，软硬件环境的故障也可能引发我们的服务异常。软硬件环境的故障包括：单机故障如硬盘坏、内存坏、网卡坏、系统死机失去响应或重启等。机房或机架故障如断网、断电等。区域性故障如运营商网络故障、DNS 服务商故障、自然灾害比如地震等。

对于一个规模化的服务系统，从不间断服务的角度，低概率的软硬件环境故障就会变成必然事件。比如我们考虑，假设一块硬盘的寿命是三年，也就是说每 1000 天可能会发生一次故障，但如果我们的服务集群有 1000 块硬盘，这就意味着平均每天都会坏一块盘。

其三，终端用户的请求也可能引发故障。比较典型的场景是秒杀类，短时间内大量的用户涌入，导致系统的承载能力超过规划，产生服务的过载。当然还有一些场景比如有针对性的恶意攻击、特定类型的用户请求导致的服务端资源大量消耗等，都可能引发服务故障。

所以，一个合理的服务治理系统，不只是需要能够及时反应业务系统的健康状况。更重要的是，要在发生了故障的情况下，能够提供故障跟踪与排查的有效线索，方便业务 SRE 可以快速定位跟踪的根因（Root Cause），并进行及时的止损。

当然，大部分情况下服务是正常的。但这并不代表我们就不会遇到麻烦。从服务单例用户的角度来说，我们服务可能没有发生故障，但是我们的某个用户就是访问不了我们的服务，或者访问服务没有得到预期的结果。

从单例用户的支持角度，我们还需要考虑服务的可支持性。为什么我访问不了？为什么我点击某个按钮没有反应或者报错？如果我们不体系化去考虑这些问题，我们的售后支持将极其低效。

综上所述，一个服务治理系统看起来是这样的：

所以虚拟机和容器技术的诞生，促使人们开始探索物理资源和应用服务之间的解耦。而一旦我们完成了这一步，服务的逻辑视图就完全语义化了，它与物理资源就只是一个应用的过

程。物理资源环境发生任何故障，都可以迅速在新的硬件设备上重新构建。

对 SRE 来说，机器的损坏和生命周期管理基本上已经不需要任何操作了。硬件已经被池化。成千上万的机器加入系统，或者出现问题，被修复，这一切都不需要 SRE 的任何操作。

这意味着，随着系统的层次结构不断上升，我们完成了从手动触发，到自动触发，到自主化。

这正是今天 DCOS（数据中心操作系统）走的路。

## 结语

今天我们对本章服务治理篇做了概要的介绍。服务治理不是纯理论，没有简洁的抽象问题模型，我们面对的是现实世界的复杂性。这些现实的复杂性，必然带来解决方案的复杂性。

直到今天为止，很多问题仍然没有被圆满解决。但是，它们的确已经在被解决的边缘。相关领域的探索与发展，日新月异。

如果你对今天的内容有什么思考与解读，欢迎给我留言，我们一起讨论。下一讲我们聊聊“事务与工程：什么是工程师思维”。

如果你觉得有所收获，也欢迎把文章分享给你的朋友。感谢你的收听，我们下期再见。

---



# 许式伟的架构课

从源头出发, 带你重新理解架构设计

许式伟  
七牛云 CEO



新版升级: 点击「 请朋友读」, 20位好友免费读, 邀请订阅更有**现金**奖励。

© 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 加餐 | 如何做HTTP服务的测试?

下一篇 48 | 事务与工程: 什么是工程师思维?

## 精选留言 (3)

写留言



Aaron Cheung

2019-10-08

数据中心操作系统后续会深入讲解吗

展开

作者回复: 会涉及



3



靠人品去赢

2019-10-08

现在不是有一个发现服务的東西, 发现服务提供服务, 服务治理会不会比以前要轻松些。还有就是docker和K8s, 我知道k8s更好, 但是又说不出哪里好, 实际上都是里面扔东

西，我自己玩的话还是喜欢docker，毕竟一大堆现成的拿来直接玩，老师怎么看待这两者，K8S到底有什么过人之处呢？

展开 ∨

作者回复: 服务发现只是服务治理中的一个小点



👍 2



**leslie**

2019-10-12

Google SRE其实换到其它行业其对应的属性个人多年DBA&&OPS的经验感觉：其实现在已经不再是Google SRE的最初解释了，就像现在的OPS要做好OPS的事情其实至少应当具备DevOps或者全栈工程师的能力，近半年一直在极客时间去学习、反思、探索，其实现在的Ops应当是以过去的Ops为主且兼备全栈的能力，就像老师课程中所说的服务治理建立自治化系统。

展开 ∨

