



下载APP



10 | 法则四：架构设计中怎么判断和利用技术趋势？

2022-01-04 郭东白

《郭东白的架构课》

课程介绍 >



讲述：郭东白

时长 29:08 大小 26.68M



你好，我是郭东白。

上节课我们讲了为什么要顺应技术的生命周期。但是“往者不可谏，来者犹可追”，我们就不能抓住一个技术萌芽和发展的机会吗？今天我们就来探讨一下这个问题。

技术未来的趋势，谁主沉浮？

领资料

你有没有想过，到底是谁决定技术的未来呢？其实大多数人都不决定技术的未来，哪怕是雷军，他也在思考该怎么顺势而为，“于万仞之上推千钧之石”。那么，技术大趋势的动力来自哪里呢？



我认为技术真正的推动力来自**市场，需求规模决定技术走向**。这由经济规律决定，不以个人意志为转移。哪怕一个头部大厂，也不能完全决定技术的未来，而是市场规模和成本结构决定技术发展的走势。

那么细分到我们关心的软件架构，它的发展来自市场的三个核心推动力，自底向上分别是：硬件技术发展、软件行业的竞争格局，以及垂直行业的商业模式的进化。越往上，距离我们越近，迭代得也越快。

从硬件技术发展看软件架构的未来

我们先从软件发展的最底层的推动力——硬件发展讲起。

看技术趋势，甚至看任何发展趋势，都要先找前置量（Leading indicator）。对于软件发展而言，**硬件的革新往往是前置量**。

首先，硬件技术进化的驱动力是需求规模。计算机硬件技术从巨型机、大型机、小型机，到 PC、Mobile 的进化过程，就是市场需求规模的增长过程。随着市场需求规模越来越大，就会有越来越多的技术创新参与到规模效应中来。

这种有规模效应的技术创新几乎是赢家通吃。而市场中最后剩下的玩家，很少会超过两个。一般是领先的玩家开发主流技术，来服务主流用户和主流场景。而略小的那个玩家，则去服务主流技术覆盖得不够好的边缘场景和小众用户。

其次，硬件技术的进化来自驱动用户侧的体验变革，再传递到软件的变革。从商用小型机到个人电脑，从 Mac 到 PC，从 iPhone 到 Android Mobile，都是某个硬件厂商先做大，然后硬件厂商对市场份额的争夺决定软件的走势。

这也是为什么我说硬件变革是软件行业的前置量。它在提醒我们，虽然我们是软件架构师，但关注硬件侧的变革很重要。顺便说一句，当下很火的元宇宙也符合这个规律。Nvidia 先做大，大量量产的 GPU 寻找增量市场，试图吹大元宇宙这个泡泡。接着 EPIC、Facebook 和 Microsoft 入场。

不过国内互联网软件研发人员，对硬件的关注普遍不够。他们受国内大玩家的影响，比如阿里、腾讯、头条和美团，大多把注意力放在了端（Mobile）上。但硬件更根本的改变是从设备（device）开始的，而不是端。

在移动互联网时代，之所以 PC、Web 玩家都去抢夺 Mobile 的端入口，并不是 Mobile 端一下子变得更有技术前景了，而是因为 Mobile 端之后的智能设备开始大规模量产了。大量的设备带来大量的新用户，也就是 Mobile 端侧的需求被迅速放大。有了大量的需求，才会有大量的软件供给争夺份额。

我之前在美国的甲骨文、微软和亚马逊都工作过，对设备的争夺感触颇深。这些公司看到了苹果在智能设备端的长期霸权和成功，所以一而再再而三地尝试做自己的设备。甲骨文的 Exadata、微软的 Xbox 和亚马逊的 Kindle，都是在各自领域比较成功的设备。尤其是 Kindle，给亚马逊带来了在电子书领域决定性的优势。

可以看到，国内很多软件玩家对硬件的关注不够，导致战略不够清晰，投入也不坚决，甚至可以说完全没有。我觉得这是个重大的失策。而国内做得最好的华为，本来最有希望做出一个现象级的产品，但因为封锁现在也不太好说了。

那么怎么关注硬件呢？在硬件生产一侧，规模效应和出货量的关系很大。其实这个关系非常普遍，几乎适用于各行各业。我特别建议你去一两家大型工厂实地参观感受一下。不必是与计算机相关的生产商，只要是现代化、规模化和流水线作业的生产环境就行。这样你很容易就能理解出货量对规模效应有多么关键了。

相比厂房，流水线上移动的单个产品，甚至是工人的成本，都可以忽略不计。一旦流水线建好了，就必须开足马力使劲生产，这样才能靠量产带来规模化的利润回收。

这也是为什么摩尔定律对英特尔公司的成功这么关键，因为他们靠摩尔定律对未来的收入和生产规模有了一个相对准确的预估。

接下来的规律就与我们软件架构师息息相关了。**计算设备的出货量决定软件架构！**因为单个计算设备的定价基本是由出货量决定的。计算设备的价格决定软件的计算成本，而软件架构必须建立在合理的计算成本之上。

过去二十年分布式计算的架构就是这个规律的一个实例。本来分布式软件开发的成本要远远高于单机软件的开发成本。但是当低成本算力和高网络带宽同时出现时，分布式计算的硬件和分布式软件开发加在一起的总成本，就远远低于单机计算和单体软件开发的总成本。而算力因为摩尔定律和尼尔森定律的缘故，在不断以指数级别扩大规模优势，所以就

驱使全社会对分布式操作系统和计算有了大量投入。这些投入的结果就是分布式架构最终胜出。

在硬件行业的竞争上几乎永远是**出货量为王**，尤其是产品价格远远高于原材料价格的半导体行业。不过靠出货量而最终竞争胜出在半导体之外的行业也适用，国内电器和电子设备制造业就是个不错的例子。

那么如果你知道摩尔定律和出货量为王的规则，你会决定怎么决定架构呢？我认为**在架构设计上，要尽量寻找利用和放大规模效应机会，保障开发软件所带来的价值在规模增长的过程中不断变大。**

我们先看你在架构决策上该如何利用好硬件的规模效应。这一点，我们可以对比一下国内的云计算厂家和亚马逊 AWS 的决策。

亚马逊 AWS 在硬件上有一个原则，就是用最低的成本采购最便宜的硬件。甚至在运行环节，比如说机房运维，机房的冷却参数也可以设置在国家安全标准所允许的最高温度，以此来追求极端的低成本。结果就是 AWS 在软件和机制层面开发了大量的高可用计算能力，使得建设在不太稳定的硬件环境上的服务，能够稳定运行，不受硬件故障的影响。最后呢，系统规模越大，采购量越大，单位硬件成本越低，高可用带来的增量价值对比硬件成本也越大，而开发成本分摊到每个计算单元上就越小。所以他们的生意越大，自己的竞争壁垒越高。

但在国内，云计算刚火起来时，大多数云供应商在采购和软件投入的决策上是反过来的。很多云厂商采购比较高端的硬件来提升系统稳定性，然后通过营销手段拉新，最后通过提价赚取利润。很少有企业投入力量做底层高可用技术的。结果是国内很多玩家入场早，但至今都做不大。

顺便说一句，在软硬件的长期战略上，我特别佩服华为。华为非常强调规模优势，愿意花大价钱在有长期规模优势的地方投入技术。有时间你可以研究一下。

另外，我认为在纯软件层面，其实也有类似硬件的出货量为王的规模效应。举个例子，前不久，我和一位在大厂做稳定性的同学聊起了 Service Mesh。他表示对比大厂的稳定性体系，Service Mesh 还是有很大差距的。大厂做了多年稳定性，场景也特殊，其他的小企业

根本达不到大厂的量级。Service Mesh 不论从成本和可靠性，都没办法和大厂的现有体系匹敌，未来也不可能。

不过我却觉得这是典型的软件规模优势必然取胜的场景。成千上万的企业必然提供为 Service Mesh 提供海量的测试案例。对比为单个顶级用户定制一次性的解决方案，一个为整个行业提供基础设施的玩家必然是赢家。我这个预测可以让时间来见证。

总结一下：软硬件都具备规模效应，最终出货量最大的玩家会以更低的价格和更好的质量赢得市场，而硬件的发展往往会左右软件架构的走向。最终软件架构必须要利用好规模效应。

从“天神打架”看技术趋势

软件行业内的竞争，则是软件架构技术发展的第二个核心推动力。

我们大多数人都不会为一个决定技术趋势的大厂工作。即便我们为这样的大厂工作，往往也不处在决策位置上。从个人角度看，这些决定技术趋势的大厂就像天神一样。民间有言：“天神打架，凡人遭殃。”言外之意就是，天神要真是打架了，你躲得越远越好。

不过在技术层面，天神打架决定了未来的技术趋势。如果有机会看到这种场面，势必要睁大眼睛看清楚，这样才能在一众凡人中脱颖而出。

天神打架的玩法其实也逃不出竞争的普遍规律，不外乎我们老祖宗讲的合纵连横。

从竞争而来的技术往往有个特征，就是后来者为了挑战先入者的霸主地位，往往在模式和技术上更开放，合作上更为广泛（Inclusive）。比如个人电脑 Apple 和 IBM PC 之间的竞争，移动领域 iOS 和 Android 的竞争，等等。

我们用云计算 AWS 和云原生基金会 CNCF 的竞争来举例吧。这个竞争还在进行当中，分析起来更有意思些。

在云计算领域，亚马逊有非常大的先发优势。它在 2006 年正式发布 AWS，比竞争对手微软发布 Azure 早了两年。事实上，微软真正投入到云计算领域的时间更晚，要到 2010 年。而谷歌真正投入到云计算要到 2012 年。所以这么长的时间，给了 AWS 足够长的市

场渗透期。这就导致在很长一段时间里，谷歌和微软都没办法缩小他们与亚马逊之间的差距。

直到 2013 年容器技术 Docker 的出现，竞争突然出现了转机。Docker 的革命性就在于它用一个轻量级的方法，完美解决了发布和运行环境的兼容性问题。任何研发人员都能以极低的成本，与另外一个研发在云环境下合作。

谷歌和微软意识到这是他们撕开 AWS 封闭式云计算环境的一个机会。所以 2014 年，[谷歌](#)不惜把自己珍藏的核武器 Kubernetes 拿出来，让大规模的云上和跨云的调度成为可能。与此同时，也开始加速在 Cloud 的投入。在这之后，谷歌和微软在云上的竞争有了明显转机。

不过革命者 Docker 只是昙花一现，它的实力根本没办法和 AWS 抗衡。毕竟桌上玩家的赌注是以百亿美金计算的，所以大家最后还是团结在了一个非常包容（Inclusive）的开发标准的周围，那就是 CNCF。

以 Kubernetes 为代表的 CNCF 的力量到底有多大呢？2014 年，Gartner 报道说 AWS 的算力，是排在它之后 14 家竞争者算力总和的 5 倍！到了 2015 年，这个倍数变成了 10。但是当 CNCF 的大联盟在这年开始形成后，微软和谷歌就开始反攻蚕食亚马逊的市场份额。

[IDG 2020 年的调查](#)说有 55% 的调查对象（组织）使用超过一个以上的云供应商。而[亚马逊](#)的公有云市场份额却从 2018 年的 68% 降低了 2021 年初的 56%。

CNCF，尤其是 Kubernetes+Docker，事实上给了云计算以开放性和互联互通性。也就是说，不是像谷歌或微软这样的单个公司在与亚马逊竞争，而是一大批的开放生态参与者联合起来与亚马逊 AWS 竞争。

结果会是什么样呢？一般来说，开放生态在长时间竞争的过程中，会胜过封闭的单个公司。这是打群架的一帮人前仆后继和一个独行侠作战的过程。过去这种竞争往往是开放的一方最终胜出，而且胜出者不一定是最开始挑战独行侠的那位。我们可以看看未来 AWS 对 CNCF 会不会是这样。

那么这对我们架构师意味着什么呢？这意味着，我们要把架构依赖在你认为的那个赢家身上。

从商业模式看技术未来

我们讲了技术的真正推动力来自于市场，而市场的一个重要变革因素就是**商业模式**。所以商业模式也决定技术的最终走向，是个前置量。

商业模式的不断更迭往往和行业密切相关，我就用生鲜行业来举例。

互联网最早入场生鲜时，是**纯线上加中央仓储的模式**。不过这个模式很快就被验证跑不通，因为生鲜的季节性很强。

在水果蔬菜到季之前，虽然利润高，但产量少、供给稀缺，所以采购是个大难题，难以做到规模化。

到季之后，线上与线下的竞争变得非常激烈。利润薄，供应链的成本占比很高，那么中央仓储的履约模式就没有优势了。

到了季节尾声，利润虽然又再次爬高，但供给质量难以保障，对用户体验的伤害较大。而且，同样因为稀缺性带来的采购挑战，导致很难做成规模。

之后生鲜进化出了**线上线下模式**，与盒马非常类似。盒马在高端社区的模式证明这是可行的。线下拉客，线上复购。但缺点是在铺开过程中找不到足够多的高端社区，而且建店时间周期长，专业人才招聘困难，扩张速度慢。

后来就有了**前置仓的模式**，比如叮咚买菜。叮咚买菜的成本结构远远优于盒马，一是没有门店，不需要像盒马一样找专业的管理与运营人才。二是前置仓面积小，所以铺开容易，扩张速度快，成本也低。但生鲜的季节性挑战依然存在，所以叮咚买菜在扩展品类到冷藏冷冻食品上下了很多功夫。

去年**社区团购模式**火遍全国。相比前置仓，这个模式的履约成本更低，因为它把履约成本和前置仓也降低了，让团长来负责团点，用户自提。拉新成本也因为团长的存在而变得更低，所以一下子有大量玩家进场争夺市场份额。

无序的竞争带来了很多新问题，比如团长变成了稀缺资源，成本也迅速蹿升。不过相比前几种模式，社区团购这种商业模式单均成本更低、扩张更简单、对管理人才需求更低、规模效应更大，也就是说**社区模式的确更先进了**。所以这个模式扩散速度就非常快，对相关技术人员的需求也非常大。

你作为一个技术人，尤其是这个行业内的技术人，那么关注商业模式的进化就至关重要了。因为这几种商业模式背后所需要的软硬件与运营技术之间的差异，实在是太大了。这也是为什么一个行业内的老玩家，很难迅速转身去追逐另一个商业模式的原因。

所以当新的模式出现后，你要提前看到这些商业模式各自的优劣势，尤其要从成本、规模效应、增速、技术增值空间等视角来看。当你发现了一个有优势的商业模式，就要立即去关注、思考和尝试做相关的技术创新。

顺便说一句，由于这些商业模式本身具有内在价值，所以它们的进化在其他行业也会被复制。最好的例子就是平台型商业模式和前置仓模式，几乎扩散到了其他各行各业。这方面的书籍很多，我就不做介绍了。

看清楚技术发展周期之后

你可能说了，研究这些东西和我有什么关系？我就是普通的架构师，知道这些趋势又能怎样呢？

关系大了！我们可以运用这些规律去选择我们的职业，选择我们的架构。把自己和团队的未来赌在哪个方向上，决定了我们在某个商业模式上要投入多少时间去学习和研究。

就以我的职业发展为例。2009 年我在美国甲骨文工作时，公司买下了 Sun Microsystems，出了 Exadata 服务器。当时市场上的销售非常好，一台 full rack 服务器价格 100 万美金。而且供不应求，以至于很长一段时间里，连我们自己的数据库研发团队都找不到测试机。

我那时虽是个基层小架构师，但对甲骨文的这个动作并不看好。因为在硬件出货量上，小型机没有任何规模优势可言。SGI 买了 Cray，SGI 不久挂了。Sun 买了 SGI，Sun 不久也挂了。所以甲骨文买 Sun，在我看来就很不明智。公司在短期内的确有了非常好的设备侧的创新，但是对比 Intel 服务器设备，Sun 的超级计算机算力只有几十倍不到，而成本却是 300 倍。

Intel 架构当时在用户市场的驱动下仍然遵循着摩尔定律，但是 Sun 只有企业市场，所以未来出货规模发展对 Sun 是不利的。我也就算准了甲骨文最终会落败。

尽管我的职业发展和股票都有保障，但我觉得必须离开，去更具备规模化的技术单元上做技术才更有前景。后来的事实也证明我的选择路径是正确的。甲骨文一度是全球最大的软件企业，但在做出这个和其他不够合理的决策后，慢慢就风光不再了。

所以作为普通架构师，研究技术趋势有一个显而易见的好处，就是**帮你选择一个正确的行业**，正所谓“男（女）怕入错行”嘛！

我做这个决策时，人才供给市场竞争还没有现在这么激烈。这更加说明，你要更早期地从硬件发展、软件技术竞争格局和商业模式进化的角度，去看未来的技术趋势和架构机会。

假设你真的看清楚了一个趋势和赛道，你该怎么做动作呢？我的建议是**你一定要把自己 100% 的身心投入到你认定的方向上**。

过去十年，在互联网领域是个赢家通吃的年代，未来很长一段时间应该还是这样。除非大多数国家在政策上做大幅调整，否则在这种大环境下，下一个小注就等于没下。无论是公司还是个人，在一个没有前途的赛道上，投入不投入都是必死。但在一个有前途的赛道上，投入了也要下大赌注才能赢（All-in）。

除此之外，这个法则对架构师的日常工作也极具指导价值。**当你评审别人的架构选型时，一定要关注他是否采用了一个已经有规模优势，或者是即将具有规模优势的技术。**

如果满足，那么这个架构建议基本就是合理的。在此基础之上，你可以再通过其他维度去看架构选型。反之，如果选择了即将失去规模优势，或者不具备规模优势的技术，你就有必要跟他探讨这么做选型是否合理。

有些技术同学喜欢展示一些小众技术。动机可能是好的，选型理由也比较充分（比如在性能或可用性维度上有优势），但从更长期的规模优势看，选型可能就不合理，你必须要阻止。

假设你还在职业生涯初期，看不清技术趋势怎么办？你是该看其他老师建议的 TCP/IP 原理、C 语言编程入门呢？还是找个火热的词搏命？

我先给你分享一个规律。**任何一个新技术，你进入的时间就决定了你的未来。**可以这么说，萌芽期赌命，增长期圈钱，至捧期交学费，灵感期拿价值，产出期养老，衰老期做死，退出期剖腹。

你要是年纪轻轻，就不要去学习任何产出期、衰老期，甚至是退出期的技术。我强烈建议你去看萌芽期的技术。光脚的不怕穿鞋的，万一赌中了呢？

什么是萌芽期的技术呢？其实最好的源头，就是从顶尖学术会议中，找一下最近三年论文增长最快的领域。如果你没有特别的偏好，那么哪怕追一个大词也行，比如元宇宙。你去这个大词的始作俑者，对元宇宙而言就是 Nvidia 那里，看看他们的开发者社区玩什么。

对于个人而言，技术的初期充满风险，但值得进入。在这个赌命的过程中，你不但可以学习设计思路，看到一个技术的成长与发展过程，而你自己也能获得先发优势。如果精力允许，那么多看些萌芽期的技术，对于你的成长而言会非常有价值。

这个时候，勇气很重要，投入到一个萌芽期的技术的确意味着更大的风险。而当你听到某个大词已经满世界流行，到处有人写书开课时，其实它已经过了至捧期。晚了！

小结

我来总结一下第四条生存法则。我们多数时间都在思考 and 解决小尺度的问题，但对我们个人发展和工作产出而言，更重要的是去看大尺度的问题，比如技术的生命周期。

在计算机时代，时间是个非常恐怖的概念。一个互联网技术的先进性和稀缺性很少有超过 3 年的。微服务、高并发就是很好的例子，七八年前还是稀缺和高增值的能力，现在已经被开源框架和云计算厂商变成了互联网的基础能力。

所以，技术的生命周期对于一个架构师而言，有一层很重要含义：**架构师需要不断监控自身能力的有效性和增量价值，不断提升自身能力的稀缺性和价值创造的空间。**而这个不断监控当下技术发展的过程，就会让你在更大尺度上的思考变得更准确了。

我们这节课讲了这么多关于技术趋势和规律的事情，就是期望你能够尊重技术规律，爱惜时间。**在计算机科学的世界里，你不能以考古学家的心态来学习和做事。**

在具体的架构设计的过程中，尤其是在企业现有人员、技术框架和竞争环境的约束下，架构师的技术选择空间会比较局限。但是哪怕在这个局限的选择空间内，你也要尽量看准趋势，选择已经有规模优势或者是即将有规模优势的技术，而不是选择接近衰老期的技术。互联网时代，搏命比抱大腿更重要！

思考题

今天的思考题比较费脑，我建议你只选一个：

1. Nvidia GPU 架构会完胜 Intel 架构吗？为什么？如果可以完胜，那它对服务端架构会有冲击吗？在这个趋势之下，你认为未来的云计算会发生什么变化？
2. 前些年国内不少大厂炒作混合部署，你认为这么做的价值在哪里？从长期来看，这是个合理的战略吗？你也可以把“合作部署”换成其他比较流行的技术来谈，比如 Event Driven Architecture、低代码、响应式编程。
3. 最近两三年里，你认为最大的硬件行业突破、软件竞争格局变化或商业模式变革是什么？这个变化会对未来的技术产生深远影响吗？为什么？

“

在架构设计中，要尽量看准趋势，选择已经有规模优势或即将有规模优势的技术，而不是选择接近衰老期的技术。互联网时代，搏命比抱大腿更重要！



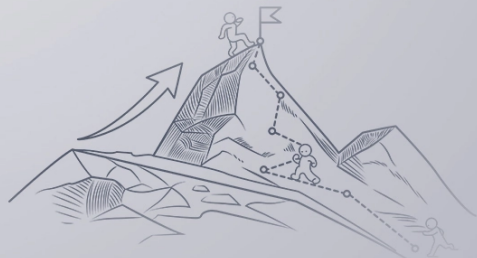
郭东白的架构课

郭东白 / 车好多集团 CTO、前阿里 P10
浙大兼职教授和博导

”



识别二维码
免费试读



注释：平台模式的相关书籍有很多，如果想看入门的，我建议可以看一下 [《平台革命》](#)。

如果今天这节课对你有帮助，欢迎你点击课程右上角的分享并赚钱按钮，把课程转发给你的同事或朋友，大家一起交流、进步。我们下节课再见！

分享给需要的人，Ta订阅后你可得 20 元现金奖励

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 09 | 法则四：为什么要顺应技术的生命周期？

下一篇 11 | 法则五：架构师为什么要关注技术体系的外部适应性？

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。