

T O P T E C H T E A M

中国顶尖 CHINA

2015年 / 第四季

技术团队

| 访 | 谈 | 录 |

InfoQ



扫一扫，了解更多

Broadview[®]
www.broadview.com.cn

用户是Docker最大的筹码

InfoQ记者采访了Rancher CEO梁胜，与他探讨了容器、OpenStack、CloudStack等技术的发展和变革。

技术人要建立自己的知识图谱

在创业公司，尤其是云服务公司，作为CTO，需要了解公司整个商业模式。



阿里崛起之路

阿里的技术架构并非一成不变，甚至在每一个阶段都会有翻天覆地的变化。

小咖秀开发仅用两天三个人

本文介绍小咖秀如何做技术积累，小团队如何玩转大用户量的应用。

浅谈“中国”语境下的公有云发展

这篇文章简要地阐述一下在“中国”这个语境下公有云发展的一些个人观点。

大师面对面：Anders Hejlsberg印象

本文为读者描述一下近距离接触大师的印象，以及Anders对TypeScript的设计理念和想法。

中国顶尖技术团队访谈录 第四季

本期主编 杜小芳

流程编辑 丁晓昀

发行人 霍泰稳

联系我们

提供反馈 feedback@cn.infoq.com

商务合作 sales@cn.infoq.com

内容合作 editors@cn.infoq.com

FACILITATING THE STUDY AND COMMUNICATION OF TECHNICIANS

让技术人学习和交流更简单

EGO EXTRA GEEKS' ORGANIZATION
NETWORKS

高端技术人员
学习型社交网络



InfoQ weue

专注中高端技术
人员的社区媒体



StuQ weue

实践驱动的
IT职业学习和服务平台



GiT GEEKBANG
INTERNATIONAL
TRAINING
极客邦培训

一线专家驱动的
企业培训服务





新浪微博
@InfoQ



微信帐号
infoqchina

InfoQ^{new}

www.infoq.com/cn



软件 正在改变世界！

InfoQ 促进软件开发领域知识与创新的传播

软件正在改变世界！InfoQ是一个在线新闻/社区网站，旨在通过促进软件开发领域知识与创新的传播，为软件开发者提供帮助。为达到这个目的，InfoQ基于实践者驱动的社区模式建立平台，提供新闻、文章、视频演讲和采访等资讯服务，所有的这一切也都是为了研发团队中那些有创新精神的人群：团队领导者、架构师、项目经理、工程总监和高级软件开发人员等。

InfoQ目前在全球有四种语言版本，分别是英文、中文、日文、葡文。

另外，InfoQ在伦敦、北京、东京、纽约、圣保罗、上海、杭州等城市举办过QCon全球软件开发大会。

InfoQ中文站将和InfoQ全球网站一样，秉承“扎根社区、服务社区、引领社区”的经营理念，与中国技术社区的专家一起，为中国软件企业和个人提供及时、高质量的技术资讯，成为连接中国企业软件技术高端社区与国际主流技术社区的桥梁。

www.qconferences.com

www.qconbeijing.com

QCon

伦敦 | 北京 | 东京 | 纽约 | 圣保罗 | 上海 | 旧金山

London · Beijing · Tokyo · New York · Sao Paulo · Shanghai · San Francisco

QCon全球软件开发大会

International Software Development Conference

QCon全球软件开发大会 (International Software Development Conference)是由InfoQ主办的全球顶级技术盛会，每年在伦敦、北京、东京、纽约、圣保罗、上海、旧金山召开。自2007年3月份在伦敦召开首次举办以来，已经有包括金融、电信、互联网、航空航天等领域的数万名架构师、项目经理、团队领导者和高级开发人员参加过QCon大会。



崔康

InfoQ总编辑，致力于中国IT领域知识与创新的传播，目前负责InfoQ整体内容的品牌和质量，同时担任QCon、ArchSummit大会的总策划。



卷首语

借着极客邦科技在旧金山举办“中国技术开放日”的机会，我在硅谷附近呆了一周，有机会实地拜访了很多知名的IT相关企业，包括Uber、Twitter、LinkedIn、RocketSpace等，也通过各种形式近距离地与当地的华人技术圈做了深入的沟通，感觉收获很大，也逐渐回答了我的问题：“我们离硅谷到底有多远？”

在硅谷期间，华人工程师问我的最多的一个问题是：“你觉得硅谷这边的华人和国内技术人员有什么区别”，老池（池

建强）在活动现场的一句玩笑话在我看来或许有些现实意义：“咱们华人工程师最大的优势在于英语好！”，的确，互联网时代的信息获取门槛不断降低，两边的工程师在知识层面上的差距正在逐渐缩小。从我了解的情况来看，硅谷的华人工程师有几处优势：

- 教育背景很好、基本功扎实，这里的华人基本上都是在美国留学然后工作，学习成绩和动手能力相对是比较好的。
- 视野比较开阔、创新意识强，硅谷

在过去几十年沉淀的创业创新精神一直在推动着整个 IT 产业的发展，在这里的技术人员也耳濡目染，所以和硅谷的工程师聊天会发现，他们的眼界比较开阔，各种新技术新想法都在关注，而且把创新当成一种习惯来做。

随着而来的是第二个问题：“我（华人工程师）应不应该回国工作和创业？”在斯坦福大学校园里我们组织的华人工程师交流会上，几位中国来的技术专家结合自身的体会讨论了很久，总结起来不外乎这样几个观点：

- 很多华人工程师的回国想法比较理想化，没有考虑家庭、孩子、环境等各方面的实际因素，往往拿了 Offer 之后犹豫不决，执行不下去。在衡量为何回国的条件中，有一个值得关注，那就是在美国的发展路径是不是已经到头了，或者已经板上钉钉没有想象的空间了。在美华人工程师，因为语言、文化等各方面的差异，一般走的都是技术路线，在管理路线上难以得志。
- 中国提供给技术人的发展机会要比美国大。中国拥有全球最大的市场之一，而且用户行为多样化、环境复杂化，同样一个应用场景在中国

就要面临很多的挑战，需要解决很多问题，这对技术人来说也是很好的机遇。这次听到一个段子，某位美国技术高管访问中国，用了一下自己的应用软件和国内的同类产品，回来跟团队说：“我们的产品比较起来简直跟 shit 一样！”目前国内的技术人待遇也有赶超欧美的趋势，不少国内 IT 企业开始从美国招募应届生，或者 CTO 级别的技术高管。

回国不一定就选择创业，各种 IT 大公司同样提供了很多的发展机会和待遇。国内的创业环境属于野蛮生长型的，攻击性比较强。从国外回来的华人在创业时比较拘谨，法律意识很强的他们在评估各种政策时普遍的反应经常是：这个不能做，那个不敢碰。对于国内的创业者来说，深谙国内政策的奥秘，通常能够找到可以伸展拳脚的机会，从而占据发展优势。但是，海外创业者也不是没有机会，在面对一个新兴领域时，特别是技术驱动的创业领域时，大家的机会是均等的，比如国内的 Docker 创业。那么，中国技术力量离硅谷到底有多远？

20 年前的 1995 年，深冬时节，在略显萧条的中关村街角处，突然树立了一块

China Tech Day

中国技术开放日



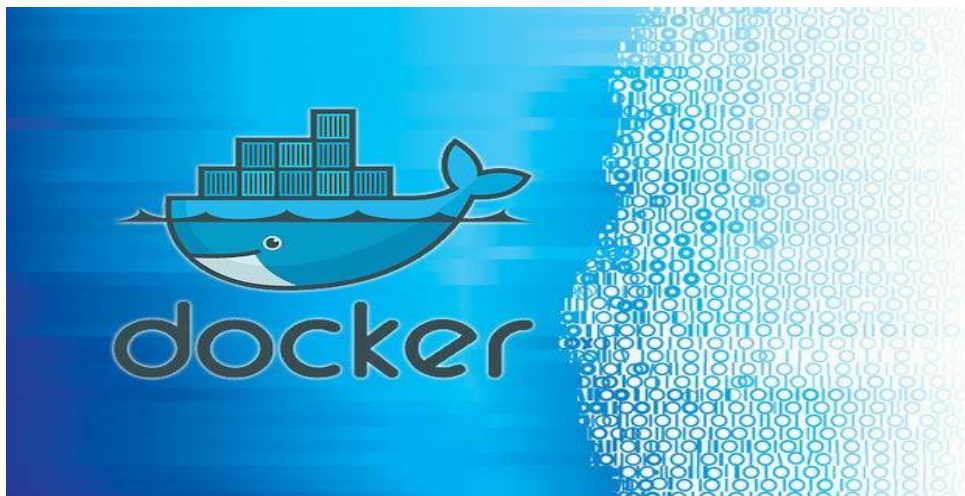
巨大的广告牌：“中国人离信息高速公路还有多远——向北 1500 米”，据说被很多人当成了真实的路标。这是当时国内最早的网络服务公司“瀛海威时空”的杰作，这块广告牌也成为中国互联网产业的标志性事件，并载入历史。

20 年后的 2015 年，深秋时节，在旧金

山市区繁华的凯悦酒店一处会议厅中，主持人正在为“ChinaTech Day 中国技术开放日”开场：“20 年来，我们一直在汲取美国的宝贵经验并自我发展，20 年后的今天，我们希望借此机会把中国的技术成果回馈给这里的朋友……”

台下随之而来的掌声就是最好的答案。

梁胜：用户是Docker最大的筹码



作者 郭蕾

Rancher Labs 是由梁胜（CloudStack 创始人）创立的一家提供容器服务的云计算公司，产品包括 RancherOS 和 Rancher。近日，Rancher Labs 和东网科技宣布在国内成立合资子公司，全面开展 Rancher 在华的业务。InfoQ 记者采访了 Rancher CEO 梁胜，与他探讨了容器、OpenStack、CloudStack 等技术的发展和变革。本文根据采访整理而成。

再谈 CloudStack 和 OpenStack

CloudStack 项目已经贡献给了 Apache，目前发展很健康。之前，大部分的开发其实都是思杰在做，而现在，社区也参与了很多功能的开发，社区这块，CloudStack 已经逐步稳定，有了自己的开发节奏。现在的基础设施云也比较成熟了，并且性能增加的速度也没有之前

那么快了，这块我觉得对用户和客户来说都非常有好处。很多的 Apache 开源项目，已经有 5 年或者 10 年的历史了，它们都很稳定，CloudStack 也差不多进入了这个状态。就目前来看，如果用户想快速搭建一个稳定的基础设施云，除了 CloudStack 以外，其实也没有太多的可以选择的技术。OpenStack 更适合一些比较大规模的项目，它和 CloudStack

的受众用户和应用场景并不完全一样。

提到 OpenStack，很多人都会关注它和 CloudStack 的对比。OpenStack 最初是由 NASA（美国国家航空航天局）和 Rackspace 合作研发并发起的，这两个机构非常开放，并且他们既不是软件厂商，又不是硬件厂商，所以 OpenStack 早期就获得了很多厂商的支持。

OpenStack 在社区和生态方面建设的确实比 CloudStack 好很多，他们做开源的思路比较特别，以厂商为最初的切入点。CloudStack 走的是另外一条路，我们想建立一个基于用户的社区，吸引更多的开发者来贡献代码，所以 CloudStack 的核心思路是简单和好用，从这方面来讲，CloudStack 就非常成功，我们也获得了很多的用户。

但从 CloudStack 和 OpenStack 的初衷来看，我觉得这两个开源软件都不是特别成功，因为当时它们都是受亚马逊的启迪，想成为企业搭建公有云或者私有云平台的基石。但现在放眼整个工业界，不管是国外还是国内的几个公有云，很少有基于 OpenStack 和 CloudStack 做的，大部分厂商反而是选择自己研发。私有云这块的市场，在我看来，并没有真正起来。就其原因，我觉得并不是技术的问题，而是产品和市场的问题。具体到美国来看，私有云市场基本上已经

等同于服务器虚拟化，大部分的公司把 OpenStack 之类的技术看作是一种自动部署虚拟机的技术。尽管 OpenStack、CloudStack 提供了 API，但很多用户并没有把云和 DevOps 流程联系到一起，像软件定义网络、软件定义存储、负载均衡之类的特性也没有真正用起来。

为什么离开思杰

简单来说，是容器技术吸引了我，我认为它是一个新兴的发展方向。特别是从私有云的角度来看，容器技术让我眼前一亮。再回来去想 CloudStack 当时碰到的一些问题，不管是市场还是产品，归根结底还是它并不能把企业的产品研发流程、DevOps 流程、大规模应用的部署流程联系起来，而容器却可以。之前 CloudStack 想拿下的客户，它们也是最早采用容器技术的团队，这也让我更坚信自己的判断。另外，对于容器技术，最吸引我的地方并不是它的效率有多高，技术有多先进，启动速度有多快之类，而是它庞大的用户群。

离开思杰后我开始做 Rancher，Rancher 想帮助客户搭建属于自己的基于容器的私有云平台。容器技术很简单，但是容器云并不简单，Rancher 想把容器云做的简单易用，以帮助企业更好的利用工程师资源和机器资源。

容器和 OpenStack 的关系

容器和 OpenStack 应该是互补的关系。因为 OpenStack 可以解决一些容器现在不能解决的问题，特别是基础设施这一层。容器需要一个效率高、安全、隔离度好的基础设施，而这正是 OpenStack 可以提供的。反过来，容器又可以帮助管理 OpenStack 之上的应用。当然，这两个技术在某些维度上又有一些功能重合，这也很正常。比如 OpenStack 有 Heat 项目可以管理上层应用，但是当应用容器化之后，用户就不需要使用 Heat 来管理了，取而代之的可能就是 Kubernetes 之类的项目。虽然功能有重叠，但基本上来说还是互补的。

社区有人也提到直接绕过 OpenStack 这样的虚拟层，直接在裸机上运行容器。就目前的情况来看，在裸机上运行容器的客户还非常少，如果这是一个趋势的话，那走到这一步还需要很久的时间。容器平台和 IaaS 的关系有点像浏览器和操作系统的关系。随着越来越多的应用可以直接运行在浏览器上，用户对操作系统的依赖越来越少。容器平台使用户更容易从一个 IaaS 云转移到另一个 IaaS 云，就像浏览器使用户更容易从一个操作系统换到另一个操作系统。但从另一个角度来看，没有传统操作系统而只能运行浏览器的电脑（像 Chromebook）

还不多见，大部分用户还是选择使用安装在传统操作系统上的浏览器。

另外，我发现在美国，DigitalOcean 也非常受用户喜欢。DigitalOcean 非常简单，直白点说就是 VPS+API，但性价比很高。仔细想想，未来的基础设施云不一定要很复杂，像 DigitalOcean 这样就足够了。如果是这样，反而对 OpenStack 有好处，因为 OpenStack 目前的市场占有率还非常小，所以它可以利用这个时机，布局好容器技术，然后简化产品，尽量做得简单易用。

如何看 Docker

Docker 这个技术很不错，并且社区也做的很棒。他们的社区基本上是 Docker 一个公司在运营，也没有采用传统的基金会模式，但我觉得他们运转地很不错，也很开放，基本上好的改进都可以被采纳。Docker 的迭代速度非常快，尽管功能不是很稳定，但可以看出这个社区的活力。如果说以 Docker 为代表的用户社区和以 OpenStack 为代表的厂商社区哪个好，我觉得还是 Docker，因为有了用户，根本不需要担忧没有厂商的支持。当然，技术和社区都是次要的，在我看来，Docker 最重要的地方还是它的用户。

Docker 这个产品给了我很大的启发，

如果让我再重新做一次 CloudStack，那我的定位一定是简单好用。之前 CloudStack 做得太重了，大而重的东西，我现在一点都不觉得好。

Rancher Labs 是一家做容器管理的公司，产品包括 RancherOS 和 Rancher，Rancher OS 是一个高效能的轻量级 Linux 分发版本，专门设计来运行容器如 Docker 等。很多人都是从 RancherOS 开始了解到 Rancher。其实为什么做 RancherOS，也和我们看好 Docker 有关系。最开始，我们并没有自己开发 OS，而是使用 CoreOS。但在使用过程中遇到了一些问题，因为 CoreOS 使用的是 Systemd 作为系统和服务的管理工具，而 Systemd 和 Docker 在某些地方配合的却不好，并且很难修改。所以在去年年底，CoreOS 就决定开发自己的容器技术 Rkt，Rkt 和 Docker 最大的区别就是它没有 Daemon，直接基于 Systemd。所以 Rkt 和 Systemd 整合的非常好，在 CoreOS 这样的系统上，Rkt 会相对更稳定。当然，从技术角度，我们也可以理解为什么 CoreOS 要这样做。

但后来我们放弃了 CoreOS，因为我们不能失去 Docker，它有那么大的用户量，这是其它容器技术没办法比的，并且 Systemd 是一个内部技术，没有太多

的真正用户。于是我们决定用 CoreOS 相反的思路重新开发一个 OS，去掉 Systemd，保留 Docker。

今年 6 月，Docker 联合其它国际大牌公司成立了 OCI，这对 Docker 来说意义重大。之前很多人总是认为 Docker 是一个厂家标准，并不是一个工业界的标准。OCI 的成立，也暗示着 Docker 公司愿意更加开放，愿意和 Linux 基金会一起推动容器技术的落地，一定程度上消除了大家的疑虑。

在容器技术飞速发展的今天，我们应怎样看待传统的私有云构建思路？在几家独大的公有云世界，是否会诞生新的挑战者？OpenStack 已发展 5 年多了，存在哪些问题？基础设施层应如何更好地支持容器化的应用？云计算和容器技术将会如何融合？云计算先驱，Rancher Labs 和 CloudStack 创始人梁胜将在 ArchSummit 全球架构师峰会中为您讲述云计算在容器时代的发展趋势。

延伸阅读：

传统企业向云演进的那些坑



极光推送黄鑫：技术人要建立自己的知识图谱



作者 陈园园

EGO是高端技术人聚集和交流的组织,每周我们都会对一位会员进行人物专访,在展示会员风采的同时,也分享会员们对技术、对工作、对人生的感悟,这次,我们邀请到了极光推送 CTO 兼首席科学家黄鑫。

扎实的底层服务是扩张关键

极光推送是一个做第三方云服务的公司,在最开始的时候做的是JPush,就是通过云服务的方式,为所有的APP提供一种推送的能力。现在,我们同时也启动了多款产品,包括聊天、支付、数据分析平台等。

我对公司的规划,是希望有很多横向的

服务,这也是为什么在Push之外,我们要做IM、Pay的原因。这样业务的横长能首先保证公司收益呈线性向上增长,但这种横向扩张的基础是必须要有一个扎实的底层服务,把你所有的业务都串起来,就好比滴滴有滴滴出行,把出租车、快车、专车、代驾等所有数据都整合到一起,彼此之间能相互利用。这种底层平台的能力是一个任何一个开

发者服务公司都必备但是却被忽略的素质。

大数据结合云产生更大价值

当把大数据和云服务结合在一起时，能产生很多不一样的价值，为开发者提供更好的服务。因为云服务汇聚的是各个 APP 开发者，如果我能把所有开发者的信息汇聚成一个大数据库，然后把这些大数据分销给不同的开发者，举个特别简单的例子，一个电商 APP，我根据大数据预测出它用户的性别、消费能力等信息，然后把这些信息提供给这个电商 APP，这样它就能据此给它的用户精准推荐商品。这个时候，云服务的能力才是真正发挥到了一个极致的地方。

我觉得这个挑战是现在国内很多云服务都没有做得很好的，并且我认为我可以去把它做得更好，这也是我加入极光推送的原因之一。

总的来说，就是底层一个公共平台做业务整合，上层很多开发者服务来保证业务的横向扩展，最上层是一个大数据平台能够让公司的业务呈一个指数型的上涨，这是我的理解里，所有云服务平台必经的一条路。

理解算法背后的世界观

我比较擅长的领域可能是数据和架构两个方面吧。大数据方面就是数据挖掘、数据分析等领域，我现在带着极光推送整个的数据团队，需要把握数据团队的技术方向，还需要做很多算法方面的研究。

对于数据挖掘，我经常提到的就是大家在理解一个算法的时候，一定要理解这个算法背后的世界观，一定要明白这个算法是为什么产生的、它解决了什么样的问题。

另外，算法是为产品服务的，所以，你关注到算法的时候，也一定要想清楚，这个算法到底能用到我们什么产品上。另外，算法是会有一个瓶颈的，所以你必须清楚什么时候需要用这个算法了，什么时候这个算法的收益率偏低，需要把它停下来了。要把算法跟商业做更好的结合，这是很多算法工程师非常欠缺的一点。

了解技术变革与技术组合

对于架构，在我看来一定要特别深入地去了解技术本身以及技术所经历过的变革。比如说 Redis，后来推出了 Redis-cluster，大家都知道 Redis-cluster 能解决哪些问题，但当我再问一个深一点的问题，从 Redis 到 Redis-cluster 为什么

经历那么久时间，推出这个新版本、做出这个新方案的时候，它有参照哪些相似的技术、有哪些优点和缺点、可能会有什么样问题。你会发现，这个问题几乎可以难倒 95% 以上的人。也就是说其实他们并没有深入地去理解一个技术的变革，这在我看来是一个很大的问题。

另外，当了解了这些技术之后，你还需要能对这些技术做出灵活的组合，因为没有任何一种技术能解决所有的问题，尤其在我们现在的时代，你的业务线非常多、你的数据非常大……所以你一定需要有这种组合技术的能力，能把若干个技术组合到一起去解决问题，这也是架构的时候非常重要的能力，对创业公司尤其如此。

曾经有一个项目，我们发现原来的技术方案不合适，后来升级的时候就用了很多技术的组合，把很多开源组件中我们认为适合的一部分都用到这个项目里，最终它的查询效率提高了将近 100 倍。当然也有以前方案的问题，但是我们还是觉得这种方案的组合其实可以产生非常大的威力。

CTO 的挑战：技术、产品与商业

我现在负责整个技术和产品研发团队的管理。大公司的 CTO 可能更像一个

Management，对具体的技术细节，他可能不需要特别了解。但对我们这样不到 200 人的创业团队来说，CTO 需要承担更多的技术职责，除了传统上的管理以及把握公司整体技术前进的方向之外，还必须要完成某些技术的攻坚。比如我会需要跟技术团队一起讨论跨 IDC 要怎么做、某一个算法要如何设计优化、甚至会讨论到某一个页面怎样的体验是最好的等等，这些很细节的层面，也都需要我作为一个 CTO 做一些把控。

当我达到现在这样程度的时候，也会对自己提出更高的挑战，因为创业公司的 CTO 其实在一定程度上决定了这个公司的技术瓶颈和技术高度，是你在引领整个公司的发展，所以你就不能拖后腿。

别人提出一个技术方案的时候，你要能够及时响应，并且能够告诉他这个方案可能会有哪些问题。而当别人没有提出的时候，你需要做的是第一时间提出新的技术方案，并且安排人去做研究和尝试。这些都是对 CTO 更大的挑战。

作为 CTO，50% 的时间做前沿技术学习，30% 的时间做对外交流，20% 的时间做竞品研究，20% 的时间做日常管理，这是我认为比较合理的时间安排，是的，做 CTO 需要花费 120% 的时间在工作上。

另外，在创业公司，尤其是在云服务公司，作为一个 CTO，你还需要去了解公司整个的商业模式。因为你比任何人都了解技术，所以你一定要知道公司以后到底要朝着哪个方向发展，比如说到底是要专注地去做推送，还是要去做数据，还是要做开发者平台，还有如何把握彼此之间的平衡等，这些都需要你去关注。你需要把技术和商业更好的结合起来，这一点可能是我最近一年最大的进步。

也许有的人的目标就是想做一个纯粹的技术专家，但对大部分人来说，你的技术其实是为产品、为商业服务的，这是我觉得很多人需要去做更多的思考、更多地走出去看业界的一点。

建立属于自己的知识体系

我的学习模式是，刚开始，我会把一门、两门技术学得非常深、非常透，并且在脑海里形成鲜明的两个知识点。然后我会不停的去接触新技术，但是每接触一个我都会跟之前的这两个技术点做比较、建立起关联，相当于形成三个点，而且这些点之间会不停地连成很多的边，显示着这些技术彼此之间的优势与劣势、相似与区别等，到最后就会在脑子里形成一张图。比如说 Python 支持

这样的一个语法，我会想那其他的语言有没有也支持这样的语法的，其他的语言不支持的时候，它到底是怎么实现这个问题的。这样你就会对比起来了，而且你会发现，你加的点越多，你对之前的东西理解得越深。

所以说技术在很大程度上是相通的，这也是我特别喜欢计算机的一个原因，就是需要你不停的去学习，然后就会觉得你学的东西原来越来越有趣。

这是我觉得非常重要的一个能力，尤其现在全栈工程师非常火，但是我面试的时候却发现，很多人他的知识面很广，但无论是他们技术的深入度还是技术点之间建立起的联系都是不够的。比如他说我会 Java、Python 还会 C++，但当我问它们三个之间有什么区别、共性、面对具体项目怎么选择的时候，可能他就没法很好的回答，只能说出一些很浅显的东西。在我看来，一个人不努力反而没那么可怕，因为他遇到感兴趣的事情时，我相信也许他就会自己努力起来了。最可怕的是一个人其实很努力，但是他从来不思考，我身边就有这样的人，他读了很多的书，可能有上百本技术相关的书，但是你会发现，他看完就看完了，不去想这些书背后的东西，也不去深入思考这些书能产生的价值，这是我觉得非常致命的一点。

中国技术力量：阿里技术崛起之路



作者 徐川

11月16日，QCon 全球软件开发大会将在美国旧金山隆重开幕，除了中国技术开放日的两位讲师之外，阿里在此次 QCon 旧金山还有两位讲师上台分享。四位讲师集体走上 QCon 旧金山，表明阿里的技术已经逐渐走出国门，得到世界技术同行的认可。那么为何阿里的技术能够得到认可，阿里的技术崛起之路是怎样的？

自我革新

阿里的技术架构并非一成不变，甚至在每一个阶段都会有翻天覆地的变化。在《淘宝技术这十年》这本书里有提到，淘宝最开始只是 LAMP 架构，后来从 PHP 切换到 Java，使用 Oracle 数据库，再到后面开始进入技术自研阶段，

使用修改过的 MySQL 来替代 Oracle，再到后来开始研发自己的服务，如 OceanBase 数据库来替代第三方的解决方案。

许多公司在规模变大后都会架构进行演进，但将基础设施一并替换的不多，给飞速行驶的汽车换轮子，这需要有极大的勇气，以及对技术研发的信心。

无线事业部资深技术总监庄卓然（花名南天）是见证并主导阿里业务架构演进的当事人之一。他于 2009 年加入阿里巴巴，先后在淘宝、天猫负责过营销、会员、交易、导购等系统的架构和研发工作，经历了淘宝从简单应用到成为数亿用户的分布式服务化业务平台的技术演进过程。他还是 2011–2013 年作为天猫双十一的技术负责人，推动了三年双十一的架构演进、稳定性改造和性能优化等研发工作。这次 QCon 旧金山他将代表阿里奔赴海外，对国外听众分享经验和心得。

激烈的架构演进和技术自研，迫使阿里的软件工程师深入到技术底层，了解技术细节，培养了一大批优秀的技术人才，而新的架构抛弃了老架构的包袱，能够轻装上阵。这些也为阿里后续的业务和规模扩张打下了坚实的基础。

ALL IN 战略

激烈的架构演进代表了一种冒险精神，而最能体现这个精神的口号，应该算“ALL IN”了。

ALL IN 口号是 2013 年 10 月阿里时任 CEO 陆兆禧提出，然后翌年春节过后马云在内部邮件强调的，当时提到的口号是 ALL IN 移动，但其实除了移动，阿里之前早已有过 ALL IN 之举。

其中最知名的要数去 IOE 化，去 IOE 这个话题经历了很多争议，伴随着各种炒作，但作为话题的中心，阿里的去 IOE 做的是实实在在的，它们在一开始对于去 IOE 就是全力以赴，当时做试点就是已经市场份额第一的淘宝，而在内部预算报告上，它们写出“淘宝 2010 年起不再购买小型机了”这样决绝的话，最后也的确实现了。

去 IOE 还有一个背景就是阿里云。阿里云于 2009 年创立，全力去 IOE，其实也是阿里全力上云，拥抱分布式计算。当去 IOE 实现，阿里在内部开始使用云计算后，对外开放的云计算服务也就准备得差不多了。

现在来看，去 IOE 与上云给阿里带来了在分布式计算领域的诸多人才，以及如 OceanBase 这样有商业化潜力的技术产品，并且在云计算越来越普及的今天帮阿里取得先机，成为国内公有云领域的领头羊。

ALL IN 移动由于时间不长我们难以看到明显的效果。但可以通过阿里的各种举措看到它的决心。比如人才的转移，上文提到的庄卓然，2013 年双十一之后转战移动研发战场，负责手机淘宝和移动基础平台的研发建设。他将和无线事业部高级技术专家徐昭一起，在 QCon

旧金山分享阿里移动亿级 App 背后的架构。另外还有资金的扶持，比如阿里百川项目推出 20 亿扶持移动创业者。

延伸阅读：

中国技术力量：中国技术开放日亮相 QCon 旧金山



拥抱开源

前面讲到阿里通过自我革新和 ALL IN 战略积累了雄厚的技术资源，但是，其它大公司也有自己的看家本领，真正让阿里区别于国内其它公司的是它的开源战略。

LVS 开源项目作者章文嵩（花名正明）加入阿里之后，开始在公司内部大力推动开源，成立了开源委员会，到现在阿里的 Github 账号下面已经拥有超过 80 个开源项目，不少项目都超过 2000 star，在行业内拥有广泛影响。值得一提的是，这些项目都是阿里自己在实践中用过的，并且相对于业界其它项目或解决方案，拥有自己的特色或者一技之长。

现在章文嵩已是阿里云首席科学家。

JStorm 就是一个很好的例子，这是一款流处理框架，淘宝在使用业界主流的 Storm 框架的时候，发现不能满足需求，于是用 Java 改写，加上更多功能并改进了其性能。而修改 Storm 之后，阿里没有忘记回馈社区，开源了 JStorm，并在最近捐给了 Apache 基金会。它的作者是阿里共享业务事业部高级技术专家封仲淹，在分布式计算以及实时计算和流处理上有多年经验，本次也将受邀参加 QCon 旧金山，向国外听众分享淘宝超大规模流处理实践，与 Uber、Facebook 的工程师同台争鸣。

开源将阿里的技术和广大的社区连接在一起，提高了阿里在技术上的声誉和认可度，同时，开源带来的开放和精益求精的风气，给阿里的技术团队带来正面的影响，为了开源出去的项目更好，工程师们更用心的打磨自己的项目，对技术研究得更加深入，对项目也更加负责。

招揽人才

阿里除了内部培养技术人才，在不同的阶段，也会引进业界知名的技术大牛。

除了上文提到的章文嵩，又如阳振坤（花名正祥），他是王选院士的学生，1999 年年仅 34 岁的他就已成为北京

大学首批“长江学者”，并曾获得国家科技进步一等奖，现为蚂蚁金服技术副总裁，是阿里分布式关系数据库 OceanBase 的初创者；还有前段时间的赵海平，Facebook 的第一位华人工程师，在 Facebook 期间，他创建了 HipHop 项目，重新编写和实现 PHP 语言，使其速度提高 5 到 6 倍，为公司节约数十亿美元，加入阿里后，重点攻克软件性能以及 Java 使用过程中遇到的技术问题；另外，去年阿里还引入了普渡大学计算机系和统计系的终身教授漆远。漆远是 Journal of Machine Learning Research 的执行编辑，曾为 ICML 领域主席，获得过美国 NSF Career award，作为蚂蚁金服技术副总裁和首席数据科学家，他致力于大规模机器学习平台的建立及其在

智能客服、芝麻信用、小额贷款等领域的应用。本次漆远也将参加中国技术开放日，在 QCon 旧金山上发表演讲。


这些技术人才的意义，不只是他们为阿里的业务做了哪些贡献，还有他们带来的一些文化和理念上的影响，他们让阿里的技术团队更加多元化，更加充满活力，同时他们本身也提高了阿里技术团队的平均水平。

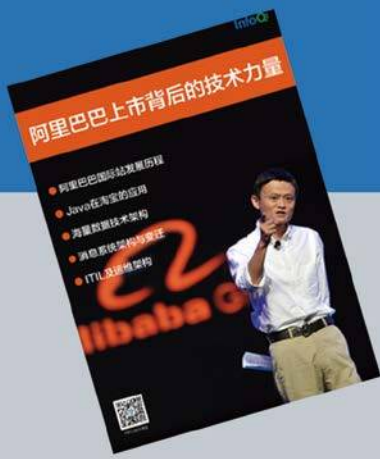
自我革新、ALL IN 战略、拥抱开源、招揽人才，通过这四个手段，阿里在国内外建立起了自己的技术口碑，而此次 QCon 旧金山，也将成为阿里技术全球征程的起点，一个新的全球技术品牌将在此扬帆起航。

免费下载 InfoQ 电子书

阿里巴巴上市背后的 技术力量

书中不仅分享了阿里技术背后的力量
还体现了技术人的梦想与坚持。







神秘的程序员们之

《如何在项目中听懂人话》

by 西乔 及神秘的技术顾问团

扫码关注
微信公众号
coderstory
原创首发



当你身处一个项目中时，仅仅掌握日常的自然语言处理能力是远不够的，必要时请参考如下的对照翻译指南，以了解地球人们真正想表达的是……

讨论进度时

现在的进度安排有些激进。
(呵呵，我们有大麻烦了)



不要紧，工作还按现在的安排继续进行。
(尼玛鬼知道问题出哪了)

(新麻烦会让人们忘掉旧麻烦)
我们将在接下来的几个迭代里弥补延误。



我给了你权力做调整。
(问责的时候就找你了)

准备发布时

编码已经完成……
(还没经过review)

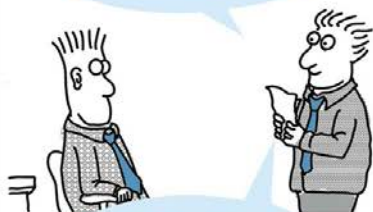
这是一次有限发布……
(不保证能用的发布)



测试被证明是主要瓶颈。
(能让某些蠢货别总找茬了吗)

讨论需求时

关于我之前说过的……
(当我之前说的都是放屁)



让我澄清一下
我们要的是什么……
(需求又改了，以前的你又都白干了)

分配资源时

有效利用核心竞争力……
(省省吧，没别的资源给你了)



开会时

现在听听我的意见：
(我级别比你高，你就别废话了)



我们达成了一致的意见……
(我说啥你照着做就行了)

这个问题我们
会后再讨论。

永远别再提这事了……

根据《项目百态》一书中相关章节改编

《神秘的程序员们》是国内首部以程序员、技术文化、项目管理及互联网创业的为主题漫画。始于09年，曾于《程序员》杂志上连载5年时间，拥有上百万读者。现于微信公众平台上重新开始发布连载，并开拓了许多关于技术文化的新分支系列。



大家预估一下这个模块的开发要多久？

又不复杂

我觉得两周肯定能弄完了

起码一个月

一看就像有坑

工程师 (新手)

小组leader

工程师 (老手)

需要大概50个工作日

留点余量

产品经理

大老板

为啥要这么长时间？
不能快点吗？
这点需求不难实现啊！

果然要多报10天 现在的员工们真是
一点进取心都没有！



客户等着用呢，加点班。
30个工作日内搞定！

我们新版本的服务一个半月后就能用了

潜在客户

销售



啊，这么慢！我们着急用啊，等不起啊。下个月能做出来不？不行我们再问问别家？

这个新版本绝对不会让您失望的，别人肯定提供不了。这样吧，我们争取一下，下个月保证拿出东西来。



两周后

进度怎么样啦？没办法，客户一个月内就要用。至少做出个demo来。

小组leader



还早着呢。

上次不是说两周就行吗？



功能倒是几天就实现出来了，但跟旧系统整合有点麻烦。

需要大概20个工作日

保险起见，
位数了说

项目经理

小组leader



需要大概40个工作日

开发们预估的时间起码得翻个倍再报上去

项目经理

产品经理



又过了两周

进度怎么样啦？
客户马上就要看产品啦

不知道啊

不就拽条
绳子嘛

一般而言，
预估开发日程的时候

.....
而等到
deadline到来的时候

以前你不是说最另一个月吗？

神秘的程序员们之

《为何受伤的总是技术人》

by 西乔 及神秘的技术顾问团

一抓到生产环境里跑就崩溃
了啊，最近的几个bug查了
四五天还没查到啊

又过了两周

为啥还没好呀，
上面天天问啊。

Shit!
还是估计
保守了啊！

TMD还在天天调界面啊！
你来调个试试？

.....

创业团队急寻
优秀技术人才或团队
高薪+股权+福利待遇！
我们比你更珍视
你的价值！
一起改变世界吧！

求贤若渴

第一周

大牛大牛，久仰久仰！
就想跟你一起干点事
兄弟，这个机会实在
是太好了，太难得了！

第二周

1个月内一定要做出
个原型来！这一轮
融资成败与非就全
看你的了！

第三周

缺人手就招，钱不是
问题，我们最重视
技术人才了。

来，招聘启事可以这样写
我们是最尊重人才的团队，
尽一切可能为有梦想的工程师们创造理想自由的环境。
不打卡、弹性工作制、免费加班晚餐；Aeron办公椅，
全套苹果设备，青轴茶轴都可以有！

第六周

你一定要带领开发团队
克服各种困难，3个月内
让产品正式上线发布！



第八周

产品这样设计我们这边实现成本太高了，以后运维起来负担也很重啊。

产品部门



你们不参与长远的战略决策，
也不懂产品和设计。让你们做成啥样就做成啥样，不要谈那么多条件好不好。

第十一周

这个需求一周的开发时间肯定是不够的。

销售部门



这关系到多少业绩、多少重要合作关系，丢了客户用户，这个责任到时候谁负？你吗？

第十五周

现在所有开发都在加班，
下一个版本发布前很多bug都得修补。

运营部门



别瞎忙了，整点有用的吧。
这个品牌推广活动老板点名了，你们配合一下。

第三十周

现在产品已经上线有一阵子了，技术部门还需要那么大规模吗？是不是能精简一下？把钱花到刀刃上。

第四十周

以后估值主要看运营和销售数字了，有些合伙人占的股份很多，又不能带来什么实际收益和增长，对公司未来发展不太有利啊。



出局

技术合伙人



然后呢？

系统崩溃

用户库数据泄漏

服务无法升级

....



(人手严重不足的)
技术部门

流量下降就是因为
你们写的服务不稳定！

为什么又要加
服务器的预算？
有个别部门只会花钱不能挣钱，
还总把事搞砸？

蒋清野：浅谈“中国”语境下的公有云发展



作者 蒋清野

作者介绍：蒋清野，曾服务于 Eucalyptus Systems Inc、Sun Microsystems Inc、北京交通大学软件学院、American GNC Corporation 等多家单位，负责多个不同领域的研发与管理工作。目前是悉尼大学信息技术学院的硕士研究生，研究领域包括开源社区，云计算市场与经济，云服务的质量、可用性与可靠性评估，以及云计算服务的互操作性。

一、公有云的规模

所谓公有云，简单地讲就是以服务的方式向公众提供计算资源。在这篇文章的范畴之内，计算资源主要指计算资源（虚拟机），但是在必要的时候会扩展到存储资源和网络资源。用各位从业人员背得滚瓜烂熟得术语来说，就是用户像用水用电一样使用计算资源，按需获取，按量计费。基于这样一个定义，一个真

正意义上的公有云需要具备一定的规模才能够达到向“公众”提供服务的基本要求。[在这篇文章的范畴之内，托管云（Managed Cloud）被认为是公有云（Public Cloud）的一种特例。]

按照 Gartner 的统计数据，在 2006 到 2014 年间，全球服务器硬件市场每年的出货量稳定在 10,000,000 台上下波动。其中，亚太地区占比在 1/4 左右，也就

是 2,500,000 台。中国境内服务器出货数量在亚太地区的占比不详，保守地按 1/5 计算也有 500,000 台。按照 3 年折旧周期估算，全国范围内现役的计算资源至少有 1,500,000 台物理服务器。作为一家服务于“中国”的产业级别的公有云服务提供商，假设其业务成熟之后拥有全国计算资源的 2%，就是 30,000 台物理服务器。再按 1:3 到 1:4 的虚拟化比例估算，则虚拟机的数量为 100,000 台左右。公有云作为一种新型服务，其市场规模尚有相当程度的自然增长空间，因此 5 年之后的公有云可能达到的规模只会比这个数字大。

根据 AWS 最近发布的财务数据，2015 年第一季度的销售收入达到 15.6 亿美元。假设来自 EC2 以及基于 EC2 的其他服务对收入的贡献占 50%，按照中等配置的 m3.large 实例（2 个 vCPU 核心，7.5GB 内存，每小时 0.14 美元）来估算，相当于 2,500,000 个 EC2 实例。根据 Rackspace 历年的财报进行估算，2014 年 Rackspace 用于公有云服务的物理服务器数量大概在 20,000 台到 30,000 台之间，换算成虚拟机也达到了 100,000 台。因此，将 100,000 台虚拟机作为一个基础目标，并非好高骛远。

基于这些估算，我们可以根据其规模判断一家公有云创业企业所处的成长阶段。

概念阶段，小于 5,000 台虚拟机。公司的终极目标相对模糊，在私有云解决方案提供商和公有云服务提供商之间摇摆不定。在战术层面，缺乏明确的技术路线图，产品形态相对原始并且没有明确的技术指标。

原型阶段，小于 10,000 台虚拟机。公司基本上将其终极目标定位为公有云服务提供商。由于公有云和私有云之间的巨大差异，必然要放弃私有云解决方案服务提供商的身份。在战术层面，基本形成相对清晰的技术路线图，基础产品（云主机）基本定型，在宕机时间和产品性能方面均有明确的技术指标。在云主机的基础上，提供能够承担中低负载的负载均衡、数据库、缓存等周边产品。

成长阶段，小于 50,000 台虚拟机。基础产品（云主机）能够满足高性能计算的要求，同时发展出一系列模块化的周边产品。普通用户完全依靠云服务提供商所提供的不同模块即可自主创建大规模可伸缩型应用（无需云服务提供商进行干预）。

成熟阶段，小于 100,000 台虚拟机。在技术方面，资源利用率开始提高，规模效应开始出现。在市场方面，客户忠诚度开始提高，马太效应开始出现。这标志着公司在公有云领域已经获得了较有份量的市场份额，其产品和技术获得了一个或者多个细分市场的广泛认可。

产业阶段，大于 100,000 台虚拟机。只有进入这一阶段，才能够认为一个服务提供商已经站稳了脚跟，可以把公有云当作一个产业来做了。至于最后能够做多，一是看国内的大环境，二是看公司自身的发展策略。

按照这样一个阶段划分，国内大部分公有云创业公司都还处于概念阶段，最多有一家创业公司已经进入原型阶段。阿里云不能够按照创业公司来看待，但是如果只统计其 ECS 部分的话，可能处于成长阶段的早期。我个人的估计，5 年后公有云拥有的计算资源可能占全国计算资源的 3% 到 5%。这意味着市场可以容纳一大一小两家进入产业阶段的公有云服务提供商，外加两到三家进入成长阶段或者成熟阶段的公有云服务提供商在一些细分市场里面深耕细作。

这也就是为什么我一直强调云计算是一片刚刚显现的蓝海。现在国内各家做公有云的公司杀得你死我活，看起来似乎已经是一片血海。在我看来，这些不过都是假象。如果一家公有云创业企业没有这样的大局观，那么我只有一个建议：“认怂服输，割肉止损，是为美德。”

二、公有云的产品

作为一个公有云服务提供商，其产品形态必然是多种多样的。但是公有云要取得成功，就不能是私有数据中心可有可

无的补充，而必须具备完全取代私有数据中心的 가능力。这意味着公有云要能够满足高性能计算的要求，普通用户完全依靠云服务提供商所提供的各种模块即可自主创建大规模、可伸缩型应用（无需云服务提供商进行干预）。12306 的查询部分迁移到阿里云勉强可以算是一个案例，问题在于这个迁移需要阿里云内部工程师的深度参与，因此不能算是一个好的案例。

鉴于产品的多样性，这里我们仅以块存储、负载均衡、自动伸缩为切入点谈谈公有云产品的特性。

块存储的磁盘 IO 指标，在从业人士当中是一个热门话题。相关讨论大都集中在云主机磁盘应该达到什么级别的 IOPS 或者是吞吐量，其实这些讨论所关注的点是完全错误的。对于公有云服务提供商来说，重要的不是云主机平均可以达到什么样的 IO 指标，而是如何根据客户的需求对整体 IO 能力进行分配。对于需要 10 个 IOPS 的低流量企业主页，为其提供 100 个 IOPS 是没有必要的。对于需要 1000 个 IOPS 的企业级应用，为其提供 100 个 IOPS 是远远不够的。套用云服务“按需获取，按量计费”的思路，IO 能力需要成为可以“按需获取，按量计费”的商品。对于需要大容量低性能的用户，可以卖存储空间；对于需要低容量高性能的用户，可以卖 IOPS。譬如说 AWS 提供三种不同

规格的 EBS 卷：传统机械硬盘 EBS 卷（magnetic）不论磁盘大小平均提供 100 个 IOPS 的 IO 能力，GP2 型 SSD EBS 卷每一 GB 保证提供 3 个 IOPS 同时又允许高达 3000 个 IOPS 的爆发性 IO，Provisioned IOPS 型 SSD EBS 卷保证可以达到用户创建该 EBS 卷时所指定的 IOPS 指标。有了这样的设计，用户可以根据其实际需求购买所需要的磁盘空间或者是 IOPS。尽管这样的购买依然受到服务提供商整体 IO 能力的限制，但是至少比所有的云主机都具备类似的“平均性能指标”要好得多。显而易见，设计这样的产品，要求云服务提供商对计算资源具有极细颗粒度的调控能力。

负载均衡也与此类似。正常的 Web 应用，其负载通常可以划分成三个档次：长期平均负载，长期高峰负载，短期爆发负载。在每秒只有数百个请求的情况下，负载均衡具备每秒处理一万个请求的能力是没有必要的。在每秒达到数万个请求的情况下，负载均衡只有每秒处理一万个请求的能力是远远不够的。如果用户按负载峰值购买负载均衡，结果是资源利用率偏低；如果用户按负载平均值购买负载均衡，结果是高峰期访问质量降低；如果用户按照实际负载切换负载均衡，结果是他再也不敢用公有云了。因此，负载均衡也要根据“按需获取，按量计费”的思路来设计，在负载降低的时候自动降级，在负载升高时自动升级。这样一种特性，就是自动伸缩。

将自动伸缩这个概念应用到云主机集群上，就是 AWS 的 AutoScaling Group（ASG）。一个 ASG 包含一组具备相同功能的云主机，应用负载降低的时候，ASG 自动杀掉多余的云主机以节省成本；应用负载升高的时候，ASG 自动启动更多的云主机以应对压力。用户按照系统的实际负载购买计算资源，既不存在处理能力不足的问题，也不存在浪费计算资源的问题。

如上几个例子，都是 AWS 在其发展早期就已经实现的技术，其核心思想都是“按需获取，按量计费”。更重要的是，通过自动伸缩这样的概念，在满足客户负载需求的前提下没有让客户花冤枉钱。我在前段时间写了一个题为“Building a scalable web application from ground zero”的入门小教程，基本上能够反映一个中型 Web 应用对计算资源的需求特征。各位做公有云的不妨对照这个教程看看类似的需求如何在自己的平台上实现。AWS 可能不是公有云的终极模式，但它至少是一种相对先进的模式，其产品对同行来说是极具启发意义的。一家公有云领域的创业公司，如果不了解、不熟悉 AWS 的产品，未免有闭门造车之嫌了。

有些人可能会说，AWS 的产品好是好，但是国内用户并不接受。这就涉及创业公司到底是想做现在的市场还是想做未来的市场的问题。如果做现在的市场，

就必须迎合市场的需求，按照客户的要求去设计产品。如果做未来的市场，就必须从技术上进行创新，指导客户按照你的思路去设计他的应用。最近几年，国内市场（尤其是互联网公司）对 AWS 所倡导的理念的接受程度是在稳步提高的。对比国际上几家公有云服务提供商，目前的局势是 AWS 一家独大，剩下几家（包括 Rackspace、Windows Azure、Google Compute Engine、HP Cloud）容量的总和与 AWS 存在接近一个数量级的差别。究其原因，在于其他几家出于种种原因没有接受 AWS 所倡导的“按需获取，按量计费”理念，只是按照传统数据中心的思路来做公有云而已。在这个大背景下，国内创业公司在熟悉 AWS 产品的基础上，模仿 AWS 的产品并争取有所创新，可能是创业早期（譬如说概念阶段）相对稳妥的发展道路。

三、公有云的成长

公有云的成长，涉及两个问题：一是用户增长，一是财务回报。

在用户增长方面，阿里云目前的方法有两个，一个是将存量用户（万网的用户，天猫的商户）往云上迁移，另外一个发展是发展政府客户。这两种客户，其特点都是对负载的要求不高（天猫整体的负载很高，但是大部分商家的独立负载并不高），对“按需获取，按量计费”的需求并不明显。换句话说，基本上是

将公有云当作传统的服务器托管的替代品来用。以阿里云目前的状况来看，将这两部分用户做好只是时间问题。从规模上看，把这两部分用户做好了，阿里云应该可以从成熟阶段进入产业阶段。问题在于，做好这两部分用户只能让阿里云拥有公有云的皮毛，并不能让阿里云拥有公有云的本质。这种情况和 Rackspace 往公有云转型过程中所遇到的问题类似。Rackspace 创立于 1998 年，以服务器租赁起家，平均每年新增服务器数量 10,000 台左右。受 AWS 的影响，Rackspace 从 2008 年起开始做公有云，但是其思路一直是用虚拟机替代物理服务器，并没有从“按需获取，按量计费”这样的思路去设计其公有云产品。仔细研究 Rackspace 从 2006 年到 2014 年间的财报数据，可以看到其收入总额和服务器数量基本上呈线性增长的趋势。换句话说，Rackspace 只是在做物理服务器的替代品，公有云部分并未对其业务产生重大影响。另外，一个值得探讨的问题是在“中国”这个语境下是否真的需要类似于 AWS 的“按需获取，按量计费”的公有云？或者说，“按需获取，按量计费”这样的需求，在所有需求中到底占多大份量。根据个人的观察，“按需获取，按量计费”这样的理念，即使是在国内互联网行业当中也还有待进一步推广，在其他行业中的接受程度显然要更低。受政策影响，未来三到五年政府在计算资源采购方面全面向公有云倾斜，而这部分用户关心的只是供应商的

名字是否有“云”字，至于这个“云”字后面是啥完全不在考虑之列。我不止一次听在政府部门做 IT 的同行说领导要求项目一定要用上阿里云，至于用阿里云干啥完全没有要求。因此，每次有朋友问我阿里云值不值得去的时候我都说阿里云的前景一片光明，如果能去的话当然要去。

按照王博士早些年的想法，阿里云还要为阿里巴巴集团提供服务。在王博士执掌阿里云的时期，阿里内部的人都觉得这是个笑话，不仅内心厌恶而且公开抵制。（关于王博士的故事，可以参考我两年前写的一篇短文“从王博士说起”。）现在章文嵩等人成为阿里云的主力，这个笑话便有了变成现实的可能性。至于这个可能性有多大，还得看阿里云后面两到三年的发展。一旦阿里云具备了为阿里巴巴集团提供服务的能力，为其他互联网企业提供服务更是不在话下。届时，阿里云可能会成为国内公有云领域毫无疑问的老大。2012 年 5 月我在第四届中国云计算大会的一个演讲上说“阿里云的技术也很好，但是在云计算产品的设计方面，还是比较业余的”，当时在从业人员当中引起了很大争议。三年过去了，如果在同行内部做一个横向比较的话，阿里云的基础产品和某些创业公司的产品相比尚存在较大差距。这个差距并非来自技术差异而是来自认知差异。换句话说，不是因为阿里云的工程师们技术水平不行，而是因为阿里云还

是没有从公有云的角度去设计产品。

与阿里云相比，创业公司基本上属于“三无”状态：没有存量用户、缺乏政府资源，尚未形成品牌。创业公司的用户增长过程，一期靠创始人的人品，二期靠技术推广，三期靠定向销售。所以创业公司的用户一般可以分成两类：某细分行业用户，其他创业公司。创业公司更有可能根据自己的发展思路对其早期用户进行教育，指导早期用户按照自己的思路和产品路线设计应用。这些投入在公司发展早期看似无用，但当客户的业务逐步增长而公有云并不成为其负载或者性能瓶颈的时候，他们就会成为公有云的长期客户和成功案例。2009 年 Netflix 全面转向 AWS 时业内几乎全是等着看笑话的，现在 Netflix 是运行在公有云上的最大型应用，同时也是 AWS 最有说服力的技术传教士。公有云帮助客户应对负载波动问题，使得客户可以聚焦在其自身业务上。客户的成功自然而然地导致消费增加，而其示范效应还会带来更多的客户。这样日积月累，方能形成一个良性循环。从资源投放的角度来看，提供“按需获取，按量计费”的能力要求云服务提供商预留部分计算资源用来应对客户的爆发性需求。云服务提供商只有到了一定的规模，才能够准确地预测客户对计算资源的需求，从而将闲置的计算资源降低到财务可以接受的比例之下。换句话说，客户成功才有公有云的成功，规模壮大才有公有云的盈利。

前两天看到陈沙克近期的一篇文章“一个做了 15 年运帷的老兵对公有云的深度剖析”，开篇就谈到 2014 年做公有云的几家创业公司是否盈利。问题在于公有云市场不是一个短期市场，而是一个未来十年尚有充分增长空间的市场。目前，中国的公有云市场尚属于发展早期，应该专注产品研发和客户教育。一家公有云创业公司如果在概念阶段就实现了盈利，这种盈利很有可能是不可持续的。在这里我想澄清一个广为流传的故事，那就是“由于其电子商务业务存在大量闲置计算资源，亚马逊想到了通过零售的方式盘活这些闲置资源，并在其基础上研发了公有云服务”。这样的故事听起来虽然合理，却是完完全全的无中生有。之所以对此进行澄清，是想说明 AWS 在其发展的早期同样会遇到客户教育、市场培养、需求预测等问题。通过接近 10 年的努力，AWS 基本上解决了这些问题，并在国际公有云市场上取得了一家独大的地位。由于缺乏历史数据，我们无从得知 AWS 是在第几年开始进入盈利状态的。但是从 S3 业务的指数增长曲线来看，AWS 不大可能在第四年（2010 年）末就实现盈利。

谈到财务回报，就不能不谈公有云的计费模式和定价策略。在“从微观经济学看云计算发展”一文中，我从微观经济学的角度分析了企业计算资源市场的供需关系。这些分析表明，和传统的服务器销售和服务器租赁业务相比，公有云

改变的不仅仅是计算资源的商业模式，它改变的是计算资源市场的供需关系。对于服务器销售和服务器租赁业务来说，客户的需求是刚性的。这意味着这客户通常是根据其业务规划购买计算资源，对计算资源的价格波动并不敏感。对于公有云业务来说，客户的需求是柔性的。这意味着这客户对计算资源的价格波动相对敏感，在价格下降时趋向于增加消费。对比 AWS 和 Rackspace，可以发现只有 AWS 呈现这个特性，Rackspace 的云计算业务并没有呈现这个特性。因此，我把客户的需求到底是刚性还是柔性作为区分虚拟机租赁和“按需获取，按量计费”的公有云的标准。如果你的客户的需求是刚性的，那么你只不过是在用传统数据中心的思路在做虚拟机租赁业务；如果你的客户的需求是柔性的，那么你就是在做“按需获取，按量计费”的公有云业务。从业务增长的角度来看，传统数据中心基本上是线性增长，而“按需获取，按量计费”的公有云业务是指数增长。

一种经济现象的出现，与其参与者的行为是密不可分的。换句话说，不能因为在 AWS 那里观察到了柔性需求，就断言在中国一定也会出现柔性需求。关于这一点，Rackspace 和 HP Cloud 恐怕深有体会，因为到目前为止他们还没有观察到柔性需求。在中国，创业公司如果沿用传统数据中心的思想来做公有云，结果只能是产品同质化市场红海化。反

之，如果围绕“按需获取，按量计费”这个理念去进行创新，开始的时候可能相对困难，但是只有坚持下去才有走进公有云这片蓝海的可能。

在外人看来，阿里云可以说是要钱有钱，要牛有牛，有战略有战术，是公众心目中的土豪型选手，唯一的缺憾在于五行缺（对云计算有深刻理解的）产品经理。依靠阿里巴巴的品牌和万网的销售能力，目前阿里云在国内的规模最大。但是从互联网行业的角度来看，阿里云的用户体验较差。很多人可能会认为阿里巴巴的技术很好，用阿里云应该比较放心。问题在于阿里巴巴并不等同于阿里云，就如同 Google 并不等同于 Google Compute Engine，微软也不等同于 Windows Azure。在互联网行业中，技术人员对青云和 UCloud 的认可度更高。虽然两者都还处于概念阶段，但是从其产品和运营来看，比较符合我对公有云的理解。这两者当中，青云看来更为激进，大有后起居上的势头。UnitedStack 由于全面拥抱 OpenStack 而广为人知，目前还在私有云解决方案提供商和公有云服务提供商这两个角色之间摇摆不定。私有云和公有云固然都很好，但是往深了做是截然不同的两个方向。创业公司需要聚焦，因此 UnitedStack 需要尽早在这两个角色之间做一个决断。如果决定往公有云服务提供商这个方向去做的话，建议抽空看看 OpenStack 外面的世界。（插播一下广告，

Rackspace 和 HP 都在用 OpenStack 来做公有云，两者都处于比较尴尬的状态。国内用 OpenStack 来做公有云的创业公司不妨思考一下，用 OpenStack 做公有云到底还缺少什么。我个人的直觉是用 OpenStack 做底子不是不行，但是光有这个底子肯定不行。）

本文最初发布于作者个人博客，经原作者授权由 InfoQ 中文站转载并分享。

延伸阅读：

私有云之殇：公有云这座大山



汤力嘉：小咖秀开发仅用两天三个人



作者 郭蕾

小咖秀是一款非常火爆的移动视频拍摄应用，据爱范儿消息，它三个月的下载量就超过了 1500 万。其 CEO 官韩坤表示，小咖秀只有两个后端工程师和一个运维人员，开发周期仅有两天，看来在这之前团队已经有了足够的技术积累。记得前不久，另外一个爆红的移动应用足记在突然来了 100 万用户之后就遇到了技术上的问题，这也是很多移动应用的短板。那小咖秀是如何做技术积累的？一个小团队是如何玩转如此大用户量的应用的？InfoQ 记者采访了小咖秀的 CTO 汤力嘉。

InfoQ：能介绍下你们团队和产品目前的情况吗？

汤力嘉：炫一下科技成立于 2011 年，目前公司 150 人，一半是技术产品。公司旗下三款产品：秒拍、小咖秀、Vitamio 多媒体框架。秒拍是微博官方的短视频拍摄应用，是最大的短视频分

享平台；小咖秀是时下最火的对嘴型拍摄应用，刚推出即占据 Apple Store 总榜第一半月之久；Vitamio 是多媒体一揽子解决方案，包括视频拍摄、美化、转码、播放等一系列 SDK，全球有 2 万款应用在使用，覆盖用户超过 5 亿。

InfoQ：小咖秀和秒拍都是视频类的应

用，针对这类的应用，你们的架构是怎样的？

汤力嘉：上传流程：应用端通过拍摄或导入等方式生成视频，经过特效加工美化，生成 MP4 文件。通过全局调度系统，确定就近的上传点。应用进行视频上传后，服务器会进行截图等操作。最后视频经过审核后，完成整个上传流程。（见图 1）

播放流程：用户打开应用或 HTML 5 页面上进行视频播放。根据不同的操作系统，选择合适的视频格式。通过全局调度系统，选择就近的质量最好的播放 CDN 进行播放。（见图 2）

InfoQ：小咖秀和秒拍这样的应用最大的挑战是什么？为了应对这些挑战，你

们做了怎么样的调整？

汤力嘉：在视频量爆发之后，我们凸显的问题是：如何快速甄选出优质视频，对优质内容进行最大可能的传播。因此我们加入很多触发规则以及监控体系，随时发现热点视频并快速消费。从产品层面上，我们也配合这些内容推出 24 小时热榜栏目，得到很多用户的喜爱。同时由于用户群体的多样化，不同用户消费和渴望看到的内容有很大的差别，为此我们也组建了数据挖掘团队，通过大数据分析，给不同用户推荐不一样的内容，完全根据用户的行为作推荐，通过个性化推荐后，视频播放量上升 30% 以上，这也是我们接下来持续优化的方向。

由于视频数量越来越多，内容的审核压

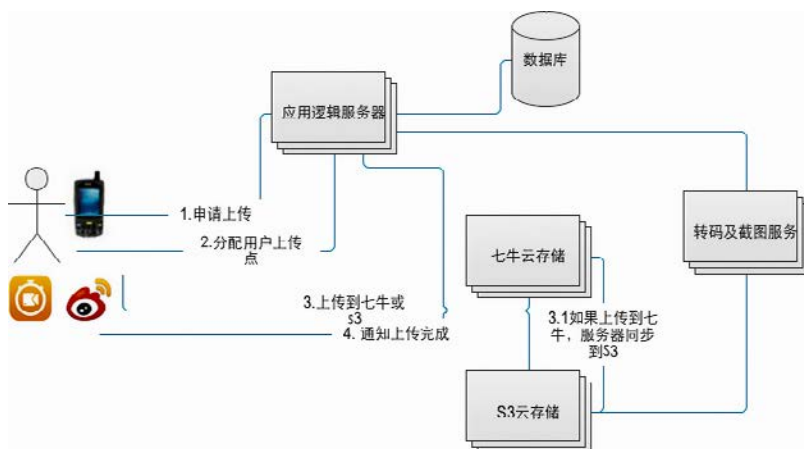


图 1

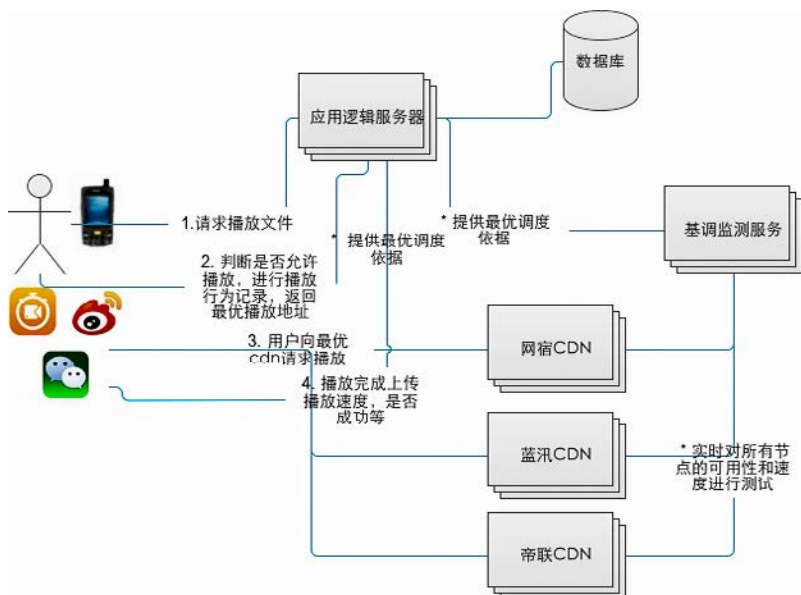


图2

力也越来越大，除了人工审核以外，我们也使用了多方面的技术和手段，进行一些半自动化的审核加速，例如对违规视频的分析 and 预审，能把高危视频优先审核；另外根据用户行为筛选高风险视频，以及应用机器学习进行反垃圾评论过滤。

InfoQ：足记在爆红之后，服务器曾一度垮掉，小咖秀有遇到这样的访问压力吗？

汤力嘉：视频分享讲究即时性，在微博等社交平台上分享尤为突出，视频在刚分享时，就已是最热的时候。例如一个明星大V的视频分享到微博上，瞬间就

会吸引他的粉丝打开观看，这时往往很考验系统对突发的海量请求做出及时响应。因此在系统的设计之初，我们就一直按着大流量突发请求的方式去做。系统的设计都是可以随时横向扩展的，所有逻辑请求都会考虑两级的缓存进行分流，确保数据库的轻负载。逻辑服务上，我们使用可弹性伸缩的云平台，以应对突发的流量。视频服务上，我们接入了多家CDN厂商进行服务，并且有实时追踪系统，能对出现故障的节点进行快速的屏蔽，同时对应每一个请求，都会选择出最优的节点进行服务（包括性价比、速度和可用性）。

InfoQ: 手机端录制的视频会比较大，特别是一些像素高的手机，小咖秀客户端有没有对视频进行压缩处理，请问使用到了哪些技术。服务器端存储的时候是如何进行优化的？

汤力嘉: 我们有强大的视频处理团队，多年来一直进行视频处理优化和应用，手机端的拍摄模块，我们使用了 Vitamio 拍摄 SDK（VCamera SDK）进行视频的拍摄和生成，视频大小会压缩到合适网络分享的程度，视频生成后，服务器端没有对视频进行二次处理，这样也加快了视频的准备过程，为视频快速传播祭奠基础。

在服务器端存储上，主要是需要考虑安全性，其次考虑视频存储的性能，尤其是视频读取的性能，便于视频的快速加载播放，所以目前的方案是在全国三地架设存储区，每一个视频都会同步三份，以确保视频的安全，同时便于就近快速访问。

InfoQ: 视频特效这块，谈谈你们的解决方案？

汤力嘉: 随着手机性能越来越好，手机对视频的处理逻辑已经越来越复杂了，我们的架构是把视频处理过程分成 source、filter、render 三个部分，source 是各种输入源，可以是文件，可以是摄

像头，可以是网络流等，用于源视频输入到处理框架上；filter 是各种各样的不同特效，因此可以很方便的做扩展；render 是处理完的视频如何输出，可以是输出成文件，可以输出到屏幕上等等。这三部分组件可以通过脚本的方式，更换不一样的处理逻辑，实现不同的特效。

InfoQ: 视频与音频的拼接,特效的处理,你们的解决方案是怎么样的？

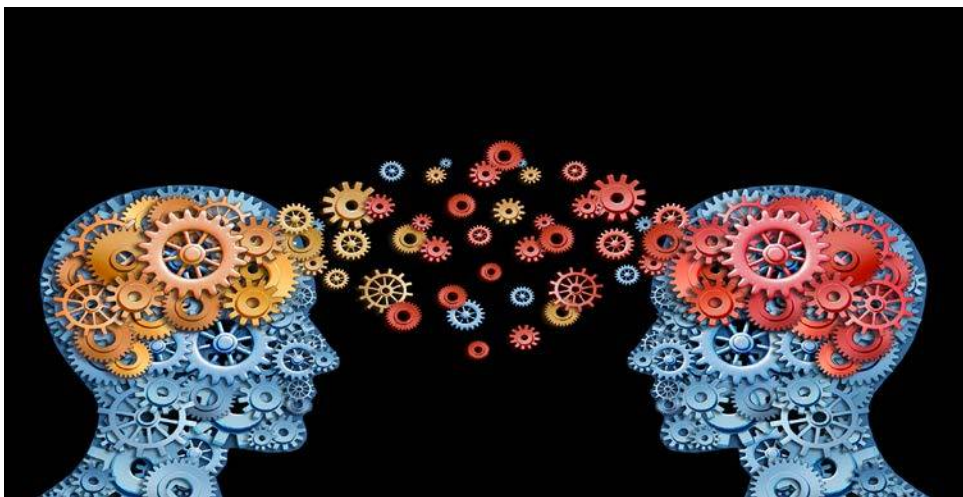
汤力嘉: 在视频的基本编辑处理方面，iOS 系统有对应的 API，可以直接使用，对于 Android 来说，我们使用 FFmpeg 框架进行基础视频编辑处理。特效的处理，我们主要使用 OpenGL 进行各种效果的叠加。

延伸阅读：

深度学习在自动问答系统中的应用



大师面对面：Anders Hejlsberg 印象



作者 徐川

Delphi、C# 之父 Anders Hejlsberg 近日首次访华，并在 10 月 24 日和 27 日参加了两场见面会，分享了他目前领导开发的 TypeScript 项目，并与国内前端开发者近距离交流。InfoQ 记者全程参与了两场见面会以及会后的采访，本文就为读者描述一下近距离接触大师的印象，以及 Anders 对 TypeScript 的设计理念和想法。

Anders Hejlsberg（安德斯·海尔斯伯格）于 1960 年出生于丹麦，在 Borland 公司开发出 Turbo Pascal 和 Delphi 之后加入微软，主持微软下一代开发工具的研发。他为微软开发和设计出 Visual J++、.NET 平台以及 C# 语言，可以说他开发出的软件和语言影响全世界整整一代程序员。目前，他是 C# 语言的首席架构师和 TypeScript 的核心开发者与

TypeScript 开源项目的重要领导人。

Anders 印象

Anders 现年 55 岁，在软件开发的道路上已经走过 30 余年，但仍然对工作充满热情。他性格和善而风趣，同时，还是个纯粹的程序员，每天仍然有很多时间花在编程上。虽然相处时间不长，但大家纷纷围着他舍不得分开，这是一位

让人愿意亲近的长者、前辈和大师。笔者将他的印象总结为三点：终身编程、开源信徒、语言大师。

终身编程

Anders 现在每天仍然花大量的时间实际写代码，并且未来会一直写下去。他很清醒的知道自己长处：“相比起管理，我还是更擅长于编程。”

事实上，Anders 的成名之作 Turbo Pascal 通过出售软件授权就让他获得了大量财富，之后微软高薪聘请更让他衣食无忧，Anders 不仅仅是可以转管理——只要他开口，微软有很多的职位在等着他——他甚至可以什么都不用做，但他仍然坚持编程，因为这是他的兴趣和激情所在。

对于他如何能达到现在的成就，他表示，1 万小时定律和兴趣是很重要的因素，编程是一项需要时间来掌握的技能，Anders 说，他回忆一下，的确是花了这么长时间才真正掌握编程，而另一方面，兴趣也是非常重要的，如果有两个工作，一个给的多些，另一个给的少些，但是是你感兴趣的事业，Anders 建议选感兴趣的那个。

国内不少资深程序员在工作一段时间后转了管理，但仍对编程难以忘怀，了解 Anders 的生平和想法，想必会有所启发。

开源信徒

Anders 对开源精神的认同和身体力行让人惊讶，他目前在 Github 上每天要提交十多次代码，很难想象他来自微软，当然，微软现在其实也在拥抱开源，并且这其中也有 Anders 的努力。

微软自 Satya Nadella（萨提亚·纳德拉）担任 CEO 后，开始融入社区，向开发者示好，其中的重要举措就是开源，Anders 是其背后的推动力量之一。不仅他之前参与的 .NET 核心引擎 CoreCLR 开源，现在的 TypeScript 也彻底是由社区主导的开源项目，微软并不干涉 TypeScript 的发展方向，所有决策都由社区得出。

在 Anders 的分享里，他总结微软现在的开源战略：

- 新开发项目首选开源；
- 创立 .NET 基金会管理开源项目；
- 开源 + 开放开发，使用并依赖 Github；
- 跨平台开发工具，典型的如 Visual Studio Code。

Anders 认为开源是软件开发的趋势，现在如果你的软件不是开源的，开发者可能更愿意选择别的开源的解决方案。

语言大师

Anders 在语言设计、开发工具、编译器等诸多领域都有非常深厚的功底，这为他设计出 C# 语言打下了基础，C# 语言被公认为设计非常优秀的语言，作为它的创造者，Anders 毫无疑问是一位语言设计的大师。在现场他分享了一些对编程语言的看法。

- **极简：**Anders 喜欢简单（simplicity），除非必要，否则不添加语言特性，一个好的语言特性是它必须的最小集。
- **设计存活时间更长的语言：**有些编程语言随着时间流逝不再流行，但有些语言如 C 语言，几十年过去了仍然被很多人使用，语言与趋势（如移动互联网、物联网）无关，语言设计者的目标是设计存活更长时间的语言。
- **困难即美味：**记者问到 Anders 对 JavaScript 语言的看法，他认为 JavaScript 在设计上有诸多问题，但仍然有其优点，更重要的是，用户很多，并且难以被替代。至于 Anders 为何会选择修补 JavaScript，他认为这对他是一项挑战，以前没有人做过，解决这一问题对他就像美味一般。

TypeScript：大 JS 生态圈的一环

TypeScript 最近发布了 1.6 版本，它在社区里的接受度也越来越高，比如有人从 CoffeeScript 迁移到 TypeScript，也有人认为它比 Facebook 的 Flow 更好。这一切都要归功于 Anders 为 TypeScript 带来的优秀设计。

Anders 分享道，其实 TypeScript 做的并不多。它的两大特性，第一个给 JavaScript 加上可选的类型系统，Anders 表示，很多事情是只有静态类型才能做的，给 JavaScript 加上静态类型后，就能将调试从运行期提前到编码期，诸如类型检查、越界检查这样的功能才能真正发挥作用。在 Anders 现场的演示可以看到，TypeScript 的开发体验的确远远超过以往纯 JavaScript 的开发体验，无需运行程序即可修复潜在 bug。

另一个特性是支持未来的 ES 6 甚至 ES 7，最近的更新都与此有关。在 TypeScript 中，你可以直接使用 ES 6 的最新特性，在编译时它会自动编译到 ES 3 或 ES 5，这个功能和最近流行的 Babel 类似，事实上，Anders 认为 TypeScript 的竞争对手并非 Dart、CoffeeScript 这类语言，而是 Babel，不过 TypeScript 的类型系统是 Babel 无法做到的，Babel 也

仅是一个编译工具，无法提供更好的开发体验。

在问到对 TypeScript 将来的规划时，Anders 显得很克制。TypeScript 不追求替代 JavaScript，没有计划直接运行在浏览器、系统里，它仅仅关注在编译到 JavaScript 之前的事情，编译之后的事情，诸如 JavaScript 性能调优等等，并不关心。

Anders 表示，TypeScript 最大的目的是让程序员更具创造性，提高生产力，它将极大的增强 JavaScript 编写应用的开发和调试环节，让 JavaScript 能够方便用于编写大型应用，进行多人协作。

这种定位让 TypeScript 能更轻松的被 JS 程序员所接受，除了上面提到社区的反应，最知名的还要数 Google 采用 TypeScript 来开发其 Angular 2 框架，Google 和微软甚至为此第一次联合召开发布会。在国内，也有 HTML5 游戏引擎 Egret 使用 TypeScript 进行开发。由此可见，TypeScript 已经足够可用，甚至进入大公司的生产环境。

当然，最后让人推荐 TypeScript 的原因，还有 Anders 本身，有如此大牛带领，相信这门语言正走在正确的道路上，它的未来不可限量。

延伸阅读：

Pixalut 优先级方法——统筹安排，优化效率





ArchSummit全球架构师峰会

International Architect Summit

全球架构师峰会（International Architect Summit，简称ArchSummit）是InfoQ关于架构垂直领域的技术峰会，邀请处于企业IT架构设计核心领域的架构师，探讨不同行业、不同领域的架构发展演变，探讨架构师们在设计和实现时的心得体会，也探讨架构师的自身成长和职业发展。每次活动我们都邀请国内外的知名技术专家来分享交流，为国内的技术人员搭建一个信息沟通的桥梁。

InfoQ 中文站 2015迷你书



顶尖技术团队访谈录 第三季

《中国顶尖技术团队访谈录 第三季》挑选的六个团队虽然都来自互联网企业，却是风格各异。希望通过这样的记录，能够让一家家品牌背后的技术人员形象更加鲜活，让更多人感受到他们的可爱与坚持。



开源启示录 第二季

我们为大家呈现此迷你书，在揭示些许开源软件规律的之外，更希望看到有更多的人和企业参与到开源软件中来。



云生态专刊 2015年04期

《云生态专刊》是InfoQ为大家推出的一个新产品，目标是“打造中国最优质的云生态媒体”。



架构师 月刊

《架构师》月刊是由InfoQ中文站针对高级技术开发和管理人员所推出的电子刊物。

EXPERTS RECOMMEND

专家推荐

▶ 程立 / CHENG LI

持续关注InfoQ好多年了。由于工作繁忙没有很多时间泡技术社区，我一直选择坚持精品与原创路线的InfoQ作为获得业界信息的主要来源。当遇到难题时也会到InfoQ上寻找灵感并常常有所收获，可以说InfoQ是我的老师、智囊和朋友，借此机会向InfoQ说声谢谢！

▶ 冯大辉 / FENG DAHUI

InfoQ，技术人都喜欢。几年下来，通过InfoQ网站获得了许多有价值的资讯，通过InfoQ的电子杂志借鉴到很多技术思路，而通过InfoQ举办的数次QCon大会，又结识了不少业界朋友。期待InfoQ坚持自己的特色，期待越办越好！

▶ 洪强宁 / HONG QIANGNING

InfoQ是我获取业内最先进的技术和理念的重要渠道。在InfoQ的帮助下，我也得以与国内外众多技术高手交流切磋，获益匪浅。感谢InfoQ！

▶ 卢旭东 / LU XUDONG

我很早就是InfoQ的注册用户了（哈哈，有好几年了吧，持续保持潜水状态），它一直是了解业界研发趋势，学习先进技术和方法的最好平台！在这里还能认识很多志同道合的朋友，InfoQ有潜力成为国内最专业、最大、最有影响力的研发社区！InfoQ的电子杂志更是必看，深浅结合，对实践很有指导性。

▶ 王文彬 / WANG WENBIN

InfoQ办的QCon大会是一个高质量的盛宴，对于最新的互联网技术和最佳实践一直在做探讨。除了邀请国内的牛人，也会有国外的大牛来做分享，对技术人员是一个不可错过的大会。

▶ 杨卫华 / YANG WEIHUA

InfoQ每年遍布全球的QCon大会是技术界的盛会，给业界很多研发方向上的启发，新浪微博的技术架构也从往届QCon大会演讲中获取了不少宝贵经验。

▶ 吴永强 / WU YONGQIANG

接触InfoQ，包括QCon，已经有好几年了，我非常喜欢它的风格，灵动、快速、实用，Moq网站、QCon、《架构师》杂志都能够紧贴互联网技术的发展前沿，带来大量的最佳实践，对我们这样发展中的公司的帮助非常大。希望InfoQ能够越做越好！

▶ 毛新生 / MAO XINSHENG

InfoQ社区是架构师的一流资讯来源，也是大家交流的桥梁。