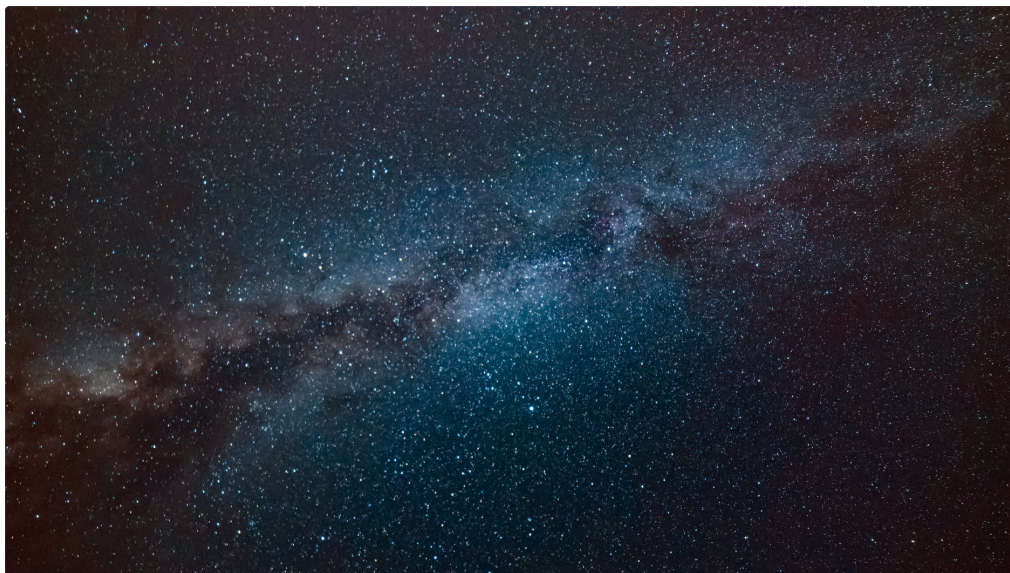


2018 年终盘点：我们处在一个什么样的技术浪潮当中？

作者：徐川

阅读量：3161 2018 年 12 月 20 日 话题：架构 云计算 云原生 方法论



2018 年接近尾声，InfoQ 策划了“解读 2018”年终技术盘点系列文章，希望能够给读者清晰地梳理出重要技术领域在这一年来的发展和变化。

正文：

最近经济寒冬的说法越来越多，身边的互联网企业裁员的也有不少，越是寒冬，我们越需要了解趋势，找准前进的方向。过去几年，互联网各种“风口”此起彼伏，到底哪些才是真正的趋势？这篇文章里我将试图分析目前互联网技术的发展，找出它们背后的原因和逻辑。

如果你长期跟进本领域的前沿技术，你会发现近十年来互联网技术发生了非常大的变化，这种变化几乎在每一个领域里发生：

- 在软件架构领域，经历了从单体应用到 SOA 再到微服务；
- 在云计算领域，经历了从虚拟机到容器；
- 在数据库领域，从关系数据库到 NoSQL 再到 NewSQL；
- 在大数据领域，从批处理到流处理；
- 在运维领域，从手工运维到 DevOps、AIOps；
- 在前端领域，从 jQuery 到 React 等三大框架；
-

除此之外，还有一些新兴的领域如 AI、区块链，从不受重视到成为显学，开启了一波又一波的风口。

单个去看这些领域的发展，会觉得纷繁杂乱没有头绪，但如果从整体上去看，会发现它们相互之间有联系，它们的发展源于一种共同的推动力，遵循着相似的逻辑。

我们当前技术浪潮的真实含义，就是我们正在走向真正的云计算时代，其它领域的发展皆由此而来，如果要更具体一点，就是：

- 云计算的技术逐渐发展成为它本来该有的模样；
- 以及与这样的云所匹配的软件架构；
- 以及与这样的架构所匹配的开发流程与方法论。

下面，我会分析几个主要的技术领域，从它们的发展历程来论述。

云计算：从虚拟化到容器到 Serverless

先从云计算说起。

2005 年亚马逊发布了 AWS，算是拉开了云计算的序幕。但是，在很长一段时间里云计算都没有兑现自己的“自动扩容、按使用付费”的宣传语。

云计算最重要的技术是分布式计算和分布式存储，分布式计算方面，最开始的技术是虚拟化，也就是所谓的“Software defined xxx”，通过对计算 / 存储和网络资源的虚拟化，同时能够给用户任意分配资源。但这里面一开始做的最好的只有文件存储这一块，AWS S3 及类似的对象存储产品给人们带来了云时代的一些实际的体验，但云服务器则还是走回了卖服务器的老路。

当然，这里的云服务器和传统服务器相比还是有优势的，至少运维不需要千里迢迢跑到机房去排查问题。但和我们想要的云服务相比还差的很远，它只是传统技术在过渡到云时代的替代品。虚拟化技术新建服务器耗时很长，在扩容方面限制很大，容器技术诞生后，才终于解决了这一问题。但现在一些 MicroVM 开始出现，比如 AWS 刚刚发布的 FireCracker，试图融合虚拟机和容器的优点，这也是当前云计算技术的一个重要关注点。

分布式存储方面，分为文件和数据库，文件通过对象存储的方式很早就解决了，数据库则面临漫长的发展过程，传统的数据库需要向分布式架构转变，同时你会发现云计算厂商成为了数据库的研发主力，这些新数据库天生就是分布式，或者天生就支持云计算特性的。

在云计算的发展过程中，有一个分支是 PaaS，最早是 2007 年推出的 Heroku，从形态上来说，它是一个 App Engine，提供应用的运行环境。PaaS 的理念被认为更贴近真正的云计算，如果你使用虚拟化的云服务器，你仍然要自己负责应用分发、部署和运维，要与各种底层接口、资源打交道，在 PaaS 上，这些都不用了，你只需要把应用上传到云端就行。

但是，之前的 PaaS 体验较差，容易造成平台绑定，难以支持大型应用，所以并没有成为主流。这些问题直到 Kubernetes 出现后才得以解决。

在 2015 年之前，OpenStack 是云计算的主流技术，很多公司包括 IBM/ 红帽都在它身上投入重注。然而，随着曾经过分天真乐观的一些公司如思科，它们试图基于 OpenStack 进入公有云市场，但在现实面前迅速败退，以及主要参与者 Nebula 的关闭，市场的信心遭遇重挫。再加上 Docker 和 Kubernetes 的快速崛起，OpenStack 的声势已经大不如前了。

然而在这么多厂商的支持下 OpenStack 是否就无敌了呢？看似紧密的社区与厂商之间的关系，在容器这个新的技术热点面前被轻松击破。厂商不再是 Pure Play OpenStack，社区贡献

2014 年 AWS 推出 Lambda 服务，Serverless 开始成为热词，从理论上说，Serverless 可以做到 NoOps、自动扩容和按使用付费，也被视为云计算的未来。但是，Serverless 本身有一些问题，比如难以解决的冷启动性能问题，因此，围绕 Serverless 的研发，以及将 Serverless 和容器技术融合也是当前的前沿课题。

Serverless 是我们过去 25 年来在 SaaS 中走的最后一步，因为我们已经渐渐将越来越多的职责交给了服务提供商。

——Joe Emison 《为什么 Serverless 比其他软件开发方法更具优势》

架构：微服务、Service Mesh 和 Serverless

云计算为应用打造了分布式的基础设施，但是，如果应用还是以传统的单体应用的思路开发，则云计算意义并不大。

这些年里，软件架构逐渐从 SOA 进化到微服务，很多人认为微服务是一种细粒度的 SOA，在去掉了 SOA 中的 ESB 之后，微服务变得更加灵活、性能更强。但是，实施微服务需要一些前提。

Martin Fowler 曾经总结过微服务实施的前提包括：

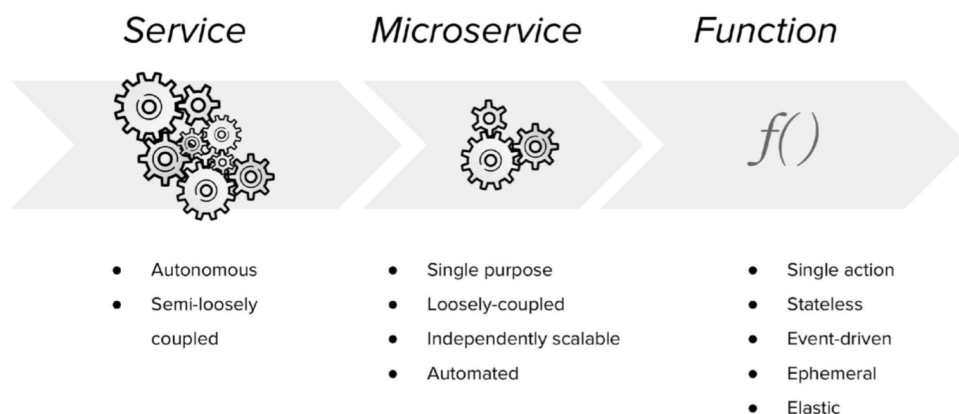
- 计算资源的快速分配
- 基本的监控
- 快速部署

这基本就是 Kubernetes 所起到的主要作用，虽然如 Spring Cloud、Dubbo 微服务框架在各方面已经非常完善，但随着云原生计算基金会的壮大，基于 Kubernetes 的微服务在社区中的热度越来越高，也开始有很多公司开始利用这一套技术栈来构建微服务。

到 2016 年，Service Mesh 开始引起社区的注意，Kubernetes 加上 Service Mesh，再加上 CNCF 的一些开源项目，基于 k8s 的微服务技术栈基本就完善了。2018 年 Istio 1.0 发布，更是为这股浪潮加了一把火，未来的微服务将是 k8s 和 Service Mesh 的天下。

Service Mesh：下一代微服务？

微服务正在逐渐走向巅峰的过程中，但它的挑战者已经出现。Serverless 或者说 FaaS 最开始只是 AWS 推出的一个功能，但随着社区和业界的跟进，逐渐有人将其认为是微服务的进化。其逻辑也很简单，从 SOA 到微服务是一个服务粒度逐渐拆分得更小的过程，FaaS 里的 Function 可以视为更小的、原子化的服务，它天然的契合微服务里面的一些理念。



<https://blog.openshift.com/wp-content/uploads/2018-04-12-Function-as-a-Service-on-Kubernetes-OCB.pdf>

(许晓斌 《从微服务到 FaaS》)

当然，关于 Serverless 如何融入到现有架构，目前并没有成熟的经验，Serverless 本身也存在一些问题，但毫无疑问这是业界关注的重点。

数据库：从 NoSQL 到 NewSQL

在过去几年，数据库的发展同样令人瞩目。

2009 年 MongoDB 开源，掀开了 NoSQL 的序幕，一时之间 NoSQL 的概念受人追捧，MongoDB 也因为其易用性迅速在社区普及。NoSQL 抛弃了传统关系数据库中的事务和数据一致性，从而在性能上取得了极大提升，并且天然支持分布式集群。

然而，不支持事务始终是 NoSQL 的痛点，让它无法在关键系统中使用。2012 年，Google 发布了 Spanner 论文，从此既支持分布式又支持事务的数据库逐渐诞生，以 TiDB、蟑螂数据库等为代表的 NewSQL 身兼传统关系数据库和 NoSQL 的优点，开始崭露头角。

从目前已有的 SQL 数据库实现方案来看，NewSQL 应该是最贴近于云数据库理念的实现。NewSQL 本身具有 SQL、ACID 和 Scale 的能力，天然就具备了云数据库的一些特点。但是，从 NewSQL 到云数据库，依然有很多需要挑战的难题，比如多租户、性能等。

——崔秋 《云时代数据库的核心特点》

本来事情发展到这里就结束了，但 2014 年亚马逊又推出一个重磅炸弹：基于新型 NVME SSD 虚拟存储层的 Aurora，它实现了完全兼容 MySQL（甚至连 bug 都兼容）的超大单机数据库，同时在性能上高出 5 倍以上。

Amazon Aurora 读后感

另外，各种不同用途的数据库也纷纷诞生并取得了较大的发展，比如用于 LBS 的地理信息数据库，用于监控和物联网的时序数据库，用于知识图谱的图数据库等。

可以说，数据库目前处于一个百花齐放的阶段，而由于云厂商的努力，基本上新的数据库都支持自动扩容、按使用付费的云计算特征。

件都是可替换的，甚至有的发生了换代。这其中，最重要的换代就是处理引擎。

最开始大数据的处理大多是离线处理，MapReduce 理念虽然好，但性能捉急，新出现的 Spark 抓住了这个机会，依靠其强大而高性能的批处理技术，顺利取代了 MapReduce，成为主流的大数据处理引擎。

随着时代的发展，实时处理的需求越来越多，虽然 Spark 推出了 Spark Streaming 以微批处理来模拟准实时的情况，但在延时上还是不尽如人意。2011 年，Twitter 的 Storm 吹响了真正流处理的号角，而 Flink 则随之发扬光大。

到现在，Flink 的目光也不再将自己仅仅视为流计算引擎，而是更为通用的处理引擎，开始正面挑战 Spark 的地位。

Apache Flink 已经被业界公认是最好的流计算引擎。然而 Flink 的计算能力不仅仅局限于做流处理。Apache Flink 的定位是一套兼具流、批、机器学习等多种计算功能的大数据引擎。在最近的一段时间，Flink 在批处理以及机器学习等诸多大数据场景都有长足的突破。

——王绍翾（大沙）《[不仅仅是流计算：Apache Flink@实践](#)》序

Hadoop 本身也遭遇了 Kubernetes 的挑战。Hadoop 本身包括专用于处理大数据的编排系统如 Yarn 等，但如 Spark/Presto/Kafka 等最重要的 Hadoop 技术已经可以在 Kubernetes 上运行，使用 Kubernetes 来运行大数据技术栈，可以更好的与其它业务集成。遭遇挑战的表现之一就是 Hadoop 技术栈的两家主要提供商，Cloudera 和 Hortonworks 最近决定合并，缓慢的增长表明市场上已经容不下两家提供商了。

[Will Kubernetes Sink the Hadoop Ship?](#)

运维：从手工运维到 DevOps

运维在过去几年遭遇了云计算技术的强烈冲击。那些依赖云计算提供商的公司，它们的运维的职责大大削弱，而自研云技术的公司里的运维则要求大大提高，过去的经验已经难以适用了。

这其中最重要的变化就是 DevOps 的出现，运维的身份职责发生了转变，它不再是专门跑任务脚本或者与机器打交道的人，而是变成了 OpenStack 或者 Kubernetes 的专家，通过搭建 / 管理相关的分布式集群，为研发提供可靠的应用运行环境。

DevOps 更重要的方面还是改变了应用交付的流程，从传统的搭火车模式走向持续交付，应用的架构和形态改变了，其方法论也随之而改变。DevOps 和持续交付也被认为是云原生应用的要素。

至于 AIOps 是 DevOps 在实践 AI 过程中的一些应用，称不上是范式的改变，AI 在运维领域还远远取代不了人的作用。

前端：前后端分离

前端在过去几年的变化同样称得上是翻天覆地，2008 年 Nodejs 的出现彻底激发了前端的生态，将 JavaScript 的疆域拓展到服务端和桌面，最终催生出大前端的概念。

如果纯粹看传统的前端开发的变化，不仅主流技术从 jQuery 转移到三大框架，更重要的是 SPA 和前后端分离的出现。

SPA 代表着前端的应用化，也就意味着胖客户端，部分业务逻辑可以从服务端转移到客户端完成。前后端分离更是将前端从后端独立出来，划定了领域边界。后端对前端来说，成为了数据层，只要接口能够正确返回

[从前后端分离看阿里 Web 应用架构演变](#)

[互联网分层架构，为啥要前后端分离？](#)

为了应对越来越大型的客户端代码，前端发展出的技术包括 TypeScript、Redux/MobX、WebAssembly、WebWorker 等，这些也是前端重点关注的技术。

AI：互联网的新基础设施

现代的 AI 是基于大数据和机器学习的，在很多公司里大数据和 AI 属于同一个数据科学的团队。在过去两年，AI 已经用各方面的成绩证明它可以成为整个互联网的基础设施之一，帮助让我们的互联网更加的智能化。

如果把 2016 年的 AlphaGo 当做现代 AI 的起点，那么 AI 发展的历史其实很短。学术界还在研究怎么提升 AI 的算法，各个公司则是急于将 AI 应用到生产环境。

AI 从感知层大致分为两大块，一块是计算机视觉，这一块已经比较成熟，无论是人脸识别、物体检测、运动检测都已经能用于实际场景中。另一块则是 NLP，虽然微软、Google 等宣称它们的 AI 翻译准确率已经极高，但实际上仍然不太好用，而多轮会话的问题没有解决，Chatbot 还是难以与人展开正常对话。

总之，真正的通用人工智能 AGI 离我们还远，至少现在还看不到头绪。AI 虽然在炒作中显得有些过热，但其技术和应用是真实的。

[致开发者：2018 年 AI 技术趋势展望](#)

[被高估的 2018：深度学习发展并没有想象的快](#)

值得注意的是，在 2018 年，国内几家涉及公有云业务的公司纷纷调整架构，将之前的云计算部门升级为智能云计算部门：

- 9 月 30 日，腾讯架构调整，新成立云与智慧产品事业群；
- 11 月 26 日，阿里集团架构调整，阿里云事业群升级为阿里云智能事业群；
- 12 月 18 日，百度调整架构，将之前的智能云事业部升级为智能云事业群。

云厂商们之所以将 AI 作为它们的顶级战略并与云计算放在一起，是因为 AI 本身需要强大的、专门定制的基础设施，是云的一个非常适合的场景；同时也因为 AI 技术有一定门槛，可以作为自身云计算差异化的一个点。总之，这些云厂商通过 AI 来卖它们的云服务。

区块链：不确定性

2018 年的区块链无疑是最有争议的话题，这里抛开那些炒作与骗局，可以看到区块链技术在 2018 年有很大的发展。

具体可分为两方面：

- 一方面是公链上一些痛点解决方案的探索和突破。包括比 POW 更好的共识机制、并发交易性能、数据存储和处理、跨链交易等等。当然，问题还远远没有得到解决。由于利益牵扯太多，这一领域也没有公认的主流解决方案。

物联网与边缘计算：为何发展不起来

物联网在过去几年一直不温不火，似乎一直在炒作中，但真正有影响力的产品和应用比较少。曾经炒过一阵的开发板最终回归为极客的玩具。物联网本身的技术，除了各种通信协议和嵌入式操作系统和开发框架之外，近两年炒的最火的就是边缘计算了，然而，边缘计算也是炒作的重灾区。

事实上，边缘计算的定义并没有清晰，甚至连边缘是什么都没有共识。有的说终端节点、智能设备是边缘，有的说 CDN 是边缘，有的说路由器、交换机是边缘，还有的说未来的 5G 基站是边缘。

边缘计算的技术目前只看到一个 EdgeX Foundry，然而在该项目里目前还看不到一个有代表性的重量级的技术，更多是一些厂商抢占风口的占位行为。

为什么会这样呢？其实好理解，因为物联网是一个很好预测的未来趋势。

从互联网到移动互联网，是一个不断扩张的过程，不但终端节点大量增加，而且每时每刻都在线，如果将这个逻辑延伸一下就是物联网了，终端从智能手机变成任何可联网的设备。

正因为这是大家都看得到的趋势，所以所有的厂商都提前在物联网布局，试图成为下一个领先者。

但互不退让的结果，就是陷入三个和尚没水吃的境地。历史上，NFC 移动支付和物联网通信协议都有这种遭遇：

- NFC 方面，在中国，银联主推 miniSD 卡的 NFC 方案，而运营商主推带 NFC 的 sim 卡，而手机厂商更愿意将 NFC 功能直接集成至手机中。在国外，美国三大运营商推出基于 NFC 的移动支付功能 Isis，苹果谷歌各自有自己的 NFC 钱包，而 Android 阵营的手机也多半将 Android Pay 功能替换为自家的支付功能。
- 物联网通信协议方面，WiFi、蓝牙、RFID、ZigBee，背后代表了不同的利益方，而在包括工业物联网等行业之后，各种私有通信协议多达数十种。

正是因为物联网协议标准的争夺，到现在我们都没有办法简单的将两个任意两个支持联网的设备相互连接，物联网无法形成像 iOS 和 Android 一样的平台。可以想象，物联网的发展还任重而道远。

智慧城市是物联网之集大成者，然而其概念从诞生到现在数十年了，我们没能看到一个成功的落地案例。

所以，物联网的发展不会像移动互联网一样一蹴而就，而是通过在共享单车上的应用，这样一个个案例积累起来逐渐进入我们的生活。

从当下的技术看未来

看了上面的盘点，你会发现云原生或者说真正的云计算是我们当下互联网技术发展的大趋势，在这个大趋势之下，推动不同的领域进行相应的发展。

其中的代表技术，就是机器学习、Kubernetes、Serverless，它们是当下这个时代技术发展的主旋律，如果你认同这个观点，你可以得出这样一个预测：



传统的应用开发将走向以容器、Serverless 为代表的真正的云计算，而随着终端和云的更深度的集成、物联网的发展、智能化的提升，云和端的界限会变得模糊，我们和理想中的互联网会更加接近。

《数字化生存》是 1996 年出版的，对于理想的互联网以前我们只是凭空的想象，而现在我们知道通过怎样的技术发展路径能抵达这个理想。

技术的本质与技术发展的逻辑

技术在不断的推陈出新，令人眼花缭乱，但如果抓住了这些技术的本质，会发现太阳底下并没有新鲜事。

如果将上面的各个领域的重要技术变革提炼一下，会发现其中的一些有共同点：

- **虚拟化**：将硬件资源虚拟为软件资源，然后进行统一调度和管理。
- **隔离**：从虚拟机到容器，再到虚拟机与容器融合，隔离的技术定义了云的形态。
- **解耦**：无论是后端的微服务、前端的前后端分离、组件化等等，都是将关注点分离，解耦合的过程。
- **编排**：大量不同的服务、任务，让他们组成一个整体，相互间能良好的协作。
- **智能化**：让服务个性化，或者让自动化替代以前需要人工完成的事情。
- **实时化**：计算和处理在极短时间内完成，从而实时的给予反馈。

当然，其中会有一些遗漏，或者有些你并不认同，但我想表达的是，这些技术存在一些共同的本质，它们是不同的领域技术发展的共同逻辑。

再进一步：是什么在推动软件的发展？

上面我们已经知道了软件的常规发展趋势，可是，如何预测软件的颠覆式创新？要预测这个，我们需要更加深入去挖掘软件进步的源头。

软件并不是凭空发展起来的，它必须要运行在各种硬件上，软件的发展，也离不开硬件的支持。

或者说，正是硬件的不断升级和变革，支撑了软件的发展进步。云计算的诞生，正是源于大型机已经无法支撑高并发，才让人们转而采用一般硬件和虚拟化、分布式的软件技术。

软件的颠覆式创新，一定是在硬件支持的基础上，随着现有的软件架构对现有硬件能力的挖掘，再发生颠覆的可能性已经较小了。

当然，这并不是说不存在，如 Docker 和比特币的诞生，都没有利用特别新的硬件能力，更多的是现有软件发展积累到一定程度的质变。

但软件创新更多的可能性，则在于硬件的颠覆上。

AWS 推出的 Aurora 数据库就是一个很好的例子，它的诞生正是基于非易失性存储技术的重大进步。现在的趋势是，硬件的创新体现在软件上的时间会越来越短。

英特尔、英伟达研发的最新芯片，也都会被云厂商第一时间订购，充分利用硬件升级带来的性能提升。

最近，还有一个新趋势是软件厂商反过来驱动硬件的进步，谷歌、阿里、华为等都开始自研用于云和终端的芯片。

如果要预测软件的发展，我们不能不去看硬件可能带来的提升，这里我们从软件运行需要的三大资源入手：

在内存领域有所突破，对于软件架构必将带来新一轮颠覆。

- **网络：**网络方面，WiFi 技术即将进入第六代，带来拥挤场合的大幅性能提升；蓝牙进入第五代，连接距离将提升至 300 米；更重要的则是 5G，相较于 4G 数百倍的数据传输速度和低至几毫米的延时，让很多应用都有了更大的想象空间。

对于技术发展的总结基本就到这里了。

选择技术是有风险的，如果是一家做 To B 或者 To C 的公司，选择了非主流的技术，只是会演变成长期的技术负债，但如果是一家面向开发者的云计算公司，选择错了技术则几乎注定了之后的衰落，无论是坚持下去还是切换成主流技术，都会因为错过最佳时机而步步艰难。这也是近年来新技术受到追捧的一个原因。

这种现象也导致了技术迭代的速度越来越快，开发者只要几年不关注新技术，就有一种被世界抛弃的错觉，于是每个人都很焦虑。

我希望用这篇文章，帮助你梳理技术的发展，知道正在发生什么，以及将会发生什么。只要知道了这些，想必不会那么焦虑了。

当然，由于个人能力所限，文章中不免有错漏之处，欢迎讨论交流。

文章版权归极客邦科技 InfoQ 所有，未经许可不得转载。

架构 云计算 云原生 方法论



26 人喜欢



收藏



评论



微信



微博



写下你的想法，一起交流

发表评论

注册/登录 InfoQ 发表评论

注册/登录

