

TOP TECH TEAM

InfoQ
ueue

中国
顶尖
技术团队
访谈录

TURING
图灵教育



扫一扫，了解更多



InfoQ

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。





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。

促进软件开发领域知识与创新的传播

架构师

3月 ARCHITECT



特别专题
探索算法和推荐系统
一次推荐系统的普及性讨论
百分点推荐引擎——从需求到架构
从路线图看PageRank算法
QCon北京 2013
参加QCon北京2013的十大理由
云端之道：网易有道搜地航专访
NoSQL的现状
深入理解Java内存模型（四）
Heroku危机带来的启示
Flash MMORPG开发中基本原则
Hadoop的现在和未来

InfoQ

每月8号出版



www.infoq.com/cn/architect

《架构师》

《架构师》是由InfoQ主办的一本主要面向架构师、高级开发人员、技术团队带头人的免费月刊电子杂志，InfoQ用户可以登录网站自由下载，目前单期月下载量近3万人次。自2009年7月创刊以来，《架构师》已经赢得了国内中高端技术人员，尤其是架构师们的认可。InfoQ的目标是将《架构师》打造成国内架构设计领域最受欢迎的技术读物。



30000



QClub——InfoQ读者俱乐部，作为InfoQ线下技术沙龙品牌，定期在全国举办非盈利的技术沙龙。邀请国内外知名公司技术总监、项目经理、高级研发工程师等走进社区，分享他们的开发实践以及对行业趋势的预测与讨论，为中国技术人员搭建交流、分享的平台，促进技术社区的发展。

目的：影响有影响力的人。

主题：围绕InfoQ中文站关注的领域设计话题。

形式：两个主题演讲 + 一个OpenSpace。

参会人员：注册InfoQ网站或参加过QCon等InfoQ旗下会议的架构师、项目经理

团队领导者、高级软件工程师等技术人员。

频率：每月都有一个以上城市举办活动。

城市：北京 上海 广州 深圳 大连 天津 西安 成都 杭州 太原 南京 武汉 福州 温州...



你应该加入 InfoQ编辑团队的 三大理由

1. 可刷脸蹭饭蹭会
2. 分享就是最好的广告
3. 跟大牛们混的多了，想不牛都难

InfoQ社区编辑永久招募中！

招聘职位：

- 强力的技术翻译
- 喜欢四处组织、参与技术活动的形象大使
- 在任意IT技术领域信息灵通的线索发现者
- 深入了解任意IT技术领域的专业内容把关人
- 擅长记笔记的新闻撰写者
- 知道如何问好问题的采访记者

我们是InfoQ编辑，我们是信息的罗宾汉。

现在就发邮件给 ada@infoq.com，
告诉我们你的专长和意向，
我们会将你培养成为一名好编辑：）



新浪微博
@InfoQ

微信帐号
infoqchina

另，InfoQ全职运营团队也需要您的支持和加盟，职位包括但不限于：技术编辑、商务编辑、销售经理、WEB前端、新媒体运营等。

工作地点：帝都。

TOP TECH TEAM

InfoQ
ueue

中国
顶尖
技术团队
访谈录



扫一扫，了解更多

目 录

卷首语	III
-----------	-----

工程篇

腾讯罗韩梅：万台规模的Docker应用实践	2
阿里巴巴毕玄：异地多活数据中心项目的来龙去脉	8
京东云首席架构师刘海锋：私有云建设的挑战与应对之道	17
华三通信研发副总裁王飓：传统通信技术与云计算的关系	29
明略数据CTO冯是聪：打造最易用的跨平台数据整合系统	34
UnitedStack创始人程辉：互联网精神+开源战略=成功的托管云	39

文化篇

云适配陈本峰：HTML5 跨屏前端框架Amaze UI的开源之道	50
豌豆荚张铎：社区是如何管理HBase项目的	58
阿里巴巴研究员赵海平：从Facebook到阿里巴巴	62

卷首语

在提笔写这篇卷首语的时候，我们刚刚为InfoQ中国过完八岁生日。回溯到2007年的3月28日，InfoQ中文站正式上线运营，从此中国的IT技术人有了一个崭新的学习和成长的平台，InfoQ中文站从一个不知名的翻译网站逐渐成长为输出全方位优质内容、对技术人有着深刻影响的媒体。几分耕耘，几分收获。让我们感到欣慰的是，InfoQ在中国IT发展的大潮中把握住了自己的方向，踏踏实实地为技术社区做了一些力所能及的事情。

很多时候，只有站在历史的峰峦之上，才能更清晰地洞察时代风云，更准确地把握前进方向。纵观过去八年的发展，中国的IT技术圈发生了翻天覆地的变化：

- 越来越多的技术领域开始引领世界。由于IT是个相对新兴的行业，特别是在一些垂直细分的领域，中国的技术水平和国外大体是在同一个起跑线上，在知识积累和人才储备方面没有传统行业差距那么大，中国没有历史包袱，再加上开放的信息渠道，所以顶尖的技术人能够在中国的技术圈里做出影响世界的成果。

- 越来越多的技术人开始追逐梦想。古语说，仓廪实而知礼节。过去八年，中国的IT行业发展很快，我们的技术人各方面有了显著的改善，除了技能和经验方面，还有软硬件的保障，越来越多的技术人在物质方面有了保障之后，开始追求兴趣，开始有了更高的理想，想做一些更有价值的事情，而这种基于兴趣和理想的技术驱动力往往很产生惊人的力量。

- 中国的技术圈越来越有吸引力。中国的巨大市场给了IT企业发展的能量，互联网、电商、社交网络等领域，都产生了国际级的大公司。中国人口多，消费能力强，潜力大，随便做点什么，就要面临海量数据的处理挑战，现在中国的一些技术挑战是世界级的，国外可能没有这样的先例和经验。这些难题对于技术人来说的吸引力是惊人的，也能获得巨大的成就感。

道路决定命运，梦想引领现实。过去八年，我们（InfoQ）把“促进软件开发领域知识与创新的传播”作为努力的目标，而未来八年，我们（极客邦）将致力于“让技术人的学习和交流更简单”。和中国的IT企业和技术人一样，我们的梦想也更高了。梦想是最令人心动的旋律，又是最引人奋进的动力。目标已经明确，我们唯有奋力前行，不负时代的使命。

《中国顶尖技术团队访谈录》是InfoQ的一个内容品牌，第一季推出之后收到了读者的热烈欢迎。这次精选了过去一年InfoQ网站上的精彩访谈内容，集结成第二季献给大家，作为InfoQ八周年的礼物，向中国的技术人致敬。

有梦想，有机会，有奋斗，一定会赢得美好未来。

InfoQ 中国总编辑 崔康

工程篇

腾讯罗韩梅：万台规模的Docker应用实践

Docker提供了一种在安全、可重复的环境中自动部署软件的方式，拉开了基于云计算平台发布产品方式的变革序幕。腾讯内部对Docker有着广泛的使用，其基于Yarn的代号为Gaia的调度平台可以同时兼容Docker和非Docker类型的应用，并提供高并发任务调度和资源管理，它具有高度可伸缩性和可靠性，能够支持MR等离线业务。为了剖析Docker on Gaia背后的实现细节，InfoQ专访了腾讯数据平台部高级工程师罗韩梅。

InfoQ：能否介绍下目前Gaia平台的状态？你们什么时候开始使用Docker的？有多大的规模？

罗韩梅：Gaia平台是腾讯数据平台部大数据平台的底层资源管理和调度系统，其上层业务包括离线、实时以及在线service服务，如Hadoop MR、Spark、Storm、Hive以及腾讯内部的Lhotse、Hermes、广点通等业务。最大单集群规模达8800台、并发资源池个数达2500个，服务于腾讯所有事业群。我们是2014年10月份正式上线Docker，之所以选择Docker，一方面是因为Gaia本来就一直在使用cgroups类型的容器，深知其在共享机器资源、灵活、轻量、易扩展、隔离等方面的重要意义。另一重要原因，是Gaia作为一个通用的云操作系统，适合所有类型的业务，但是各个业务的环境依赖是一个比较困扰用户的问题，因此我们引入Docker来解决，主要目的还是通过Docker来将Gaia云平台以更有效的方式呈献给各个业务。

我们使用的OS是腾讯内部的tlinux 1.2版本，最新版本正在tlinux2.0上测试，除了

Docker，也使用了etcd来服务注册和服务发现。我们的集群都是同时兼容Docker应用和非Docker类型的应用的，MR等应用还是使用的cgroups类型的容器，其它服务使用的Docker容器，目前，大概有15000多常驻的Docker容器，还有大量业务接入测试中。由于我们原本就是使用的cgroups容器，所以换成Docker后，性能基本也无损耗，可以满足线上需求。

InfoQ：腾讯是如何把Yarn与Docker深度结合的？

罗韩梅：在腾讯的场景下，首先一个特点就是，业务总类极大，尤其是离线处理规模很大，因此Yarn原生的调度器，效率远远跟不上，因此我们开发了自研调度器SFair，解决了调度器效率和扩展性问题。另外，腾讯的业务特性多样，因此我们引入了Docker，虽然Yarn支持不同的应用类型可以实现不同的AM（应用管理器），但是对于绝大多数应用来说，他们并不熟悉Yarn，实现一个支持容灾、可扩展的完善AM，困难较大，因此我们抽象了可以使用Docker的业务，对其进行封装，实现了统一的AM，并且对用户透明，而对用户提供的是另一套全新的基本的、易于理解的高级接口。同时，我们为Docker业务实现了统一的服务注册和发现机制，并也将其封装在了新接口中。另外，在资源管理方面，我们修改了内存管理机制，引入了磁盘和网络带宽管理。

除了Yarn之外，其实我们对Docker本身也做了一定的修改和bug修复，对于Registry等服务也做了优化，保证了其服务的高可靠和性能。

实现方面，我们并没有使用社区提供的Docker调度器，我们研发Gaia的时候社区还没有相应的调度器，并且我们也有特殊要求，需要同时支持同时支持Docker类型应用和非Docker类型应用。

InfoQ：你们如何确定哪些业务适合使用 Docker？

罗韩梅：我们认为，Docker提供了一种在安全、可重复的环境中自动部署软件的方式，拉开了基于云计算平台发布产品方式的变革序幕，因此，其实Docker对于Gaia来讲，只是一个选择，我们并不主动向业务推广Docker，而是Docker on Gaia的整套方案，所以，我们对于需要共享资源、降低成本，需要支持快速的动态扩容缩容、容灾容错，以及大规模分布式系统尤为建议使用Docker on Gaia。

InfoQ：能否详细介绍下你们对 Docker 以及 Registry 做了哪些优化？

罗韩梅：对于Docker，我们主要做了三个方面的优化：首先是bug修复，比如Docker非0退出时rm不生效，对于bindmount为true时config path无法清除等bug。其次是优化Docker的资源管理策略，比如内存的Hardlimit的管理策略，不但使用户进程容易被kill，更加造成了资源的浪费，对用户估计自己业务的资源需求也非常高。Gaia引入了EMC（Elastic Memory Control）的弹性内存管理机制。最后一个方面是资源管理纬度，Docker在资源管理纬度方面只有CPU和内存两个维度，这对于共享的云环境下需要完善，也是目前相对于虚拟机不足的地方。Gaia引入磁盘容量管理，网络出入带宽控制以及磁盘IO的控制维护。其实不仅仅在Docker层做控制，还将会引进调度器，不但实现资源的隔离，还要实现资源的保证。

对于 Registry 的优化，主要有下面几个方面：

1. 容量问题。开源的Registry是单机模式，其容量会受单个机器的限制。我们修改存储driver，取缔原有的mount方式，开发后端存储driver，直接使用HDFS，实现了存储的无限容量。

2. 可靠性和可用性的问题。单机版本的 Docker Registry，其可靠性和可用性都成了最大的问题，我们引入数据平台部的 tPG 系统，实现 Registry server 的无状态化，便于实现服务的高可用性。
3. 性能问题。将单机版的 Registry 扩展成 Registry 集群，并实现在 Registry server pool 中的负载均衡，提升性能。
4. 网络问题。解决了全国不同 IDC 的 Gaia 集群对 Registry 的访问，采取就近访问的原则，不产生跨 IDC 流量。
5. 自动同步官方镜像。Docker 提供的官方镜像中，有很多还是非常有价值的，而官方的 Registry 又在墙外，为此，我们自动同步 docker 的官方镜像到我们的私有仓库中。

InfoQ：能否介绍下目前的一个 workflow？

罗韩梅：目前使用 Docker on Gaia 的方式有三种：1) 通过 Gaia Portal；2) 直接调用 Gaia api；3) 通过上层各种 PaaS 平台透明间接使用 Gaia。比如在第一种方式中，用户通过 Gaia Portal 提交应用，之后 Gaia 调度器会自动分配资源，并且部署、启动 Docker 容器，用户可以在 Portal 上直接查看每个实例的状态、日志、异常等，甚至可以直接通过 webshell 登陆。同时，也可以根据需求对应用进行扩容、缩容、重启，以及灰度变更、停止实例/应用等操作。

InfoQ：目前平台主要部署了哪些服务？服务之间的调度是如何实现的？

罗韩梅：目前平台上的服务有 Hermes、通用推荐、广点通、游戏云等服务，很多服务都需要多实例部署，因此跨主机部署非常普遍，而不同服务直接也经常会有调用的



需求，主要是通过 Gaia 提供的服务注册和服务发现机制。具体地，NM（Yarn 的一个组件）在启动 Docker 容器时，会将该 Docker 的真正地址，包括 ip 和所有的端口映射，都会通过 etcd 做自动的服务注册。对于 Docker 内部的服务，我们通过修改 Docker 源码，扩展了几个 Gaia 相关的环境变量，将 IP 以及端口映射传入。服务的注册和发现本质上一种名字服务，因此不难理解为什么在创建应用的时候，让用户填一个应用 name 的字段。而这种基于名字的服务是贯穿这个 Gaia 的过程的：在提交作业时，用户不需要指定 Gaia master 的地址，而是通过指定 Gaia 集群的 name 即可；在获取应用的地址时，也是通过应用的名字获取；本质上 port mapping 也是一种名字，只不过是将用户原来 expose 的端口作为 name，将实际端口作为 value。至此，不难理解为什么 name 需要检查冲突。

InfoQ：万台规模的 Docker 容器，网络问题是如何解决的？

罗韩梅：网卡及交换链路的带宽资源是有限的。如果某个作业不受限制产生过量的网络流量，必然会挤占其它作业的网络带宽和响应时延。因此 Gaia 将网络和 CPU、内存一样，作为一种资源维度纳入统一管理。业务在提交应用时指定自己的网络 IO 需求，我们使用 TC（Traffic Control）+ cgroups 实现网络出带宽控制，通过修改内核，增加网络入带宽的控制。具体的控制目标有：

1. 在某个 cgroup 网络繁忙时，能保证其设定配额不会被其他 cgroup 挤占；
2. 在某个 cgroup 没有用满其配额时，其他 cgroup 可以自动使用其空闲的部分带宽；
3. 在多个 cgroup 分享其他 cgroup 的空闲带宽时，优先级高的优先；优先级相同时，配额大的占用多，配额小的占用少；
4. 尽量减少为了流控而主动丢包。

受访者介绍

罗韩梅，腾讯数据平台部高级工程师，任数据中心资源调度组副组长。2009年加入腾讯，主要从事统一资源管理调度平台的开发和运营，承担过腾讯自研云平台“台风”中Torca资源调度子系统的研发，目前主要专注于开源技术、分布式数据仓库、分布式资源调度平台、Docker等领域。

阿里巴巴毕玄： 异地多活数据中心项目的来龙去脉

大数据时代，数据中心的异地容灾变得非常重要。在去年双十一之前，阿里巴巴上线了数据中心异地双活项目。InfoQ就该项目采访了阿里巴巴的林昊（花名毕玄）。

InfoQ：首先请介绍一下数据中心异地多活这个项目。

8

中国顶尖技术团队访谈录

毕玄：这个项目在我们内部的另外一个名字叫做单元化，双活是它的第二个阶段，多活是第三个阶段。所以我们把这个项目分成三年来实现。所谓异地多活，故名思义，就是在不同地点的数据中心起多个我们的交易，并且每个地点的那个交易都是用来支撑流量的。

InfoQ：当时为什么要做这件事呢？

毕玄：其实我们大概在2009还是2010年左右的时候，就开始尝试在异地去做一个数据中心，把我们的业务放过去。更早之前，我们做过同城，就是在同一个城市建多个数据中心，应用部署在多个数据中心里面。同城的好处就是，如果同城的任何一个机房挂掉了，另外一个机房都可以接管。

做到这个以后，我们就在想，异地是不是也能做到这样？

整个业界传统的做法，异地是用来做一个冷备份的，等这边另外一个城市全部挂掉

了，才会切过去。我们当时也是按照这个方式去尝试的，尝试了一年左右，我们觉得冷备的方向对我们来讲有两个问题：第一个问题是成本太高。我们需要备份全站，而整个阿里巴巴网站，包括淘宝、天猫、聚划算等等，所有加起来，是一个非常大的量，备份成本非常高。第二个问题是，冷备并不是一直在跑流量的，所以有个问题，一旦真的出问题了，未必敢切过去。因为不知道切过去到底能不能起来，而且整个冷备恢复起来要花多长时间，也不敢保证。因此在尝试之后，我们发现这个方向对我们而言并不好。

那为什么我们最后下定决心去做异地多活呢？

最关键的原因是，我们在 2013 年左右碰到了几个问题。首先，阿里巴巴的机房不仅仅给电商用，我们有电商，有物流，有云，有大数据，所有业务共用机房。随着各种业务规模的不断增长，单个城市可能很难容纳下我们，所以我们面临的问题是，一定要去不同的城市建设我们的数据中心。其次是我们的伸缩规模的问题。整个淘宝的量，交易量不断攀升，每年的双十一大家都看到了，增长非常快。而我们的架构更多还是在 2009 年以前确定的一套架构，要不断的加机器，这套架构会面临风险。

如果能够做到异地部署，就可以把伸缩规模缩小。虽然原来就是一套巨大的分布式应用，但是其实可以认为是一个集群，然后不断的加机器。但是在这种情况下，随着不断加机器，最终在整个分布式体系中，一定会有一个点是会出现风险的，会再次到达瓶颈，那时就无法解决了。

这两个问题让我们下定决心去做异地多活这个项目。

为什么我们之前那么纠结呢？因为整个业界还没有可供参考的异地多活实现，这就意味着我们必须完全靠自己摸索。而且相对来讲，它的风险以及周期可能也是比较大的。

InfoQ：这个项目具体是怎样部署的？

毕玄：以去年双十一为例，当时我们在杭州有一个数据中心，在另外一个城市还有

个数据中心，一共是两个，分别承担 50% 用户的流量。就是有 50% 的用户会进入杭州，另外 50% 会进入到另外一个城市。当用户进入以后，比如说在淘宝上看商品，浏览商品，搜索、下单、放进购物车等等操作，还包括写数据库，就都是在所进入的那个数据中心中完成的，而不需要跨数据中心。

InfoQ：这样的优势是？

毕玄：异地部署，从整个业界的做法上来讲，主要有几家公司，比如 Google、Facebook，这两家是比较典型的，Google 做到了全球多个数据中心，都是多活的。但是 Google 具体怎么做的，也没有多少人了解。另外一家就是 Facebook，我们相对更了解一些，Facebook 在做多个数据中心时，比如说美国和欧洲两个数据中心，确实都在支撑流量。但是欧洲的用户有可能需要访问美国的数据中心，当出现这种状况时，整个用户体验不是很好。

国内的情况，我们知道的像银行，还有其他一些行业，倾向于做异地灾备。银行一般都会有两地，一个地方是热点，另一个地方是冷备。当遇到故障时，就没有办法做出决定，到底要不要切到灾备数据中心，他们会碰到我们以前摸索时所面对的问题，就是不确定切换过程到底要多久，灾备中心到底多久才能把流量接管起来。而且接管以后，整个功能是不是都正常，也可能无法确定。

刚才也提到过，冷备的话，我们要备份全站，成本是非常高的。

如果每个地点都是活的，这些数据中心就可以实时承担流量，任何一点出问题，都可以直接切掉，由另外一点直接接管。相对冷备而言，这是一套可以运行的模式，而且风险非常低。

InfoQ：不过这样的话，平时要预留出很多流量才能保证？

毕玄：没错。因为在异地或同城建多个数据中心时，建设过程中一定都会保有一定冗余量。因为要考虑其他数据中心出现故障时加以接管。不过随着数据中心建设的增多，这个成本是可以控制的。如果有两个异地的数据中心，冗余量可能是一倍，因为要接管全量。但是如果有三个数据中心，互为备份，就不需要冗余两倍了。

InfoQ：这个项目挑战还是比较大的。您可以介绍一下研发过程中遇到的挑战吗？又是怎样克服的？

毕玄：对于我们来讲，异地项目最大的挑战是延时。跨城市一定会有延时的问题。在中国范围内，延时可能在一百毫秒以内。

看起来单次好像没什么，但是像淘宝是个很大的分布式系统，一次页面的展现，背后的交互次数可能在一两百次。如果这一两百次全部是跨城市做的，整个响应时间会增加很多，所以延时带来的挑战非常大。

在解决挑战的过程中，我们会面临更细节的一些问题。怎样降低延时的影响，我们能想到的最简单、最好的办法，就是让操作全部在同一机房内完成，那就不存在延时的挑战了。所以最关键的问题，就是怎样让所有操作在一个机房内完成。这就是单元化。

为什么叫单元化，而没有叫其他名字呢？原因在于，要在异地部署我们的网站，首先要做一个决定。比如说，冷备是把整个站全部备过去，这样可以确保可以全部接管。但是多活的情况下，要考虑成本，所以不能部署全站。

整个淘宝的业务非常丰富，其实有很多非交易类型的应用，这些功能可能大家平时用的不算很多。但对我们来讲，又是不能缺失的。这部分流量可能相对很小。如果这些应用也全国到处部署，冗余量就太大了。所以我们需要在异地部署的是流量会爆发式增长的，流量很大的那部分。虽然有冗余，但是因为流量会爆发式增长，成本比较好平衡。

异地部署，我们要在成本之间找到一个平衡点。所以我们决定在异地只部署跟买家交易相关的核心业务，确保一个买家在淘宝上浏览商品，一直到买完东西的全过程都可以完成。

其他一些功能就会缺失，所以我们在异地部署的并非全站，而是一组业务，这组业务就成为单元。比如说我们现在做的就是交易单元。

这时淘宝会面临一个比Google、Facebook等公司更大的一个挑战。像Facebook目前做的全球化数据中心，因为Facebook更多的是用户和用户之间发消息，属于社交领域。但淘宝是电商领域，对数据的一致性要求非常高，延时要求也非常高。

还有个更大的挑战，那就是淘宝的数据。如果要把用户操作封闭在一个单元内完成，最关键的是数据。跟冷备相比，异地多活最大的风险在于，它的数据会同时在多个地方写，冷备则不存在数据会写错的问题。如果多个地方在写同一行数据，那就没有办法判断哪条数据是正确的。在某个点，必须确保单行的数据在一个地方写，绝对不能在多个地方写。

为了做到这一点，必须确定数据的维度。如果数据只有一个维度，就像Facebook的数据，可以认为只有一个纬度，就是用户。但是像淘宝，如果要在淘宝上买一个东西，除了用户本身的信息以外，还会看到所有商品的数据、所有卖家的数据，面对的是买家、卖家和商品三个维度。这时就必须做出一个选择，到底用哪个维度作为我们唯一的一个封闭的维度。

单元化时，走向异地的就是买家的核心链路，所以我们选择了买家这个维度。但是这样自然会带来一个问题，因为我们有三个维度的数据，当操作卖家商品数据时，就无法封闭了，因为这时一定会出现需要集中到一个点去写的现象。

从我们的角度看，目前实现整个单元化项目最大的几个难点是：

第一个是路由的一致性。因为我们是按买家维度来切分数据的，就是数据会封闭在不同的单元里。这时就要确保，这个买家相关的数据在写的时候，一定是要写在那个单元里，而不能在另外一个单元，否则就会出现同一行数据在两个地方写的现象。所

以这时一定要保证，一个用户进到淘宝，要通过一个路由规则来决定这个用户去哪里。这个用户进来以后，到了前端的一组 Web 页面，而 Web 页面背后还要访问很多后端服务，服务又要访问数据库，所以最关键的是要保障这个用户从进来一直到访问服务，到访问数据库，全链路的路由规则都是完全一致的。如果说某个用户本来应该进 A 城市的数据中心，但是却因为路由错误，进入了 B 城市，那看到的数据就是错的了。造成的结果，可能是用户看到的购买列表是空的，这是不能接受的。所以如何保障路由的一致性，非常关键。

第二个是挑战是数据的延时问题。因为是异地部署，我们需要同步卖家的数据、商品的数据。如果同步的延时太长，就会影响用户体验。我们能接受的范围是有限的，如果是 10 秒、30 秒，用户就会感知到。比如说卖家改了一个价格，改了一个库存，而用户隔了很久才看到，这对买家和卖家都是伤害。所以我们能接受的延时必须要做到一秒内，即在全国的范围内，都必须做到一秒内把数据同步完。当时所有的开源方案，或者公开的方案，包括 MySQL 自己的主备等，其实都不可能做到一秒内，所以数据延时是我们当时面临的第二个挑战。

第三个挑战，在所有的异地项目中，虽然冷备成本很高，多活可以降低成本，但是为什么大家更喜欢冷备，而不喜欢多活呢？因为数据的正确性很难保证。数据在多点同时写的时候，一定不能写错。因为数据故障跟业务故障还不一样，跟应用层故障不一样。如果应用出故障了，可能就是用户不能访问。但是如果数据写错了，对用户来说，就彻底乱了。而且这个故障是无法恢复的，因为无法确定到底那里写的是对的。所以在所有的异地多活项目中，最重要的是保障某个点写进去的数据一定是正确的。这是最大的挑战，也是我们在设计整个方案中的第一原则。业务这一层出故障我们可以接受，但是不能接受数据故障。

还有一个挑战是数据的一致性。多个单元之间一定会有数据同步。一方面，每个单元都需要卖家的数据、商品的数据；另一方面，我们的单元不是全量业务，那一定会有业务需要这个单元，比如说买家在这个单元下了一笔定单，而其他业务有可能也是需要

这笔数据，否则可能操作不了，所以需要同步该数据。所以怎样确保每个单元之间的商品、卖家的数据是一致的，然后买家数据中心和单元是一致的，这是非常关键的。

这几个挑战可能是整个异地多活项目中最复杂的。另外还有一点，淘宝目前还是一个高速发展的业务，在这样的过程中，去做一次比较纯技术的改造，怎样确保对业务的影响最小，也是一个挑战。

InfoQ：要将延时控制在 1 秒之内，网络和硬件方面都有哪些工作？

毕玄：如果网络带宽质量不好，1秒是不可能做到的。我们在每个城市的数据中心之间，会以一个点做成自己的骨干网，所以可以保障不同城市之间的网络质量。但是要保证到1秒，还必须自己再去做东西来实现数据的同步，这个很关键。这个东西现在也在阿里云上有开放了。

InfoQ：异地多活其实也是实现高可用。阿里技术保障的梁耀斌（花名追源）老师会在4月23日~25日的QCon北京2015大会上分享《你的网站是高可用的吗？》，因为当时的题目和内容也是您参与拟定的。您可以先谈一下其中的一些标准吗？

毕玄：其实每家比较大的互联网公司，每年可能都会对外公开说，我们今年的可用性做到了多少，比如4个9或者5个9。但是每家公司对可用性的定义可能并不一样。比如说，有的公司可能认为业务响应时间超过多少才叫可用性损失，而其他公司可能认为业务受损多少就是可用性损失。

我们希望大家以后有一个统一的定义，这样就比较好比较了。我们发现，真正所有做到高可用的网站，最重要的一点是故障恢复时间的控制。因为出故障是不可避免的，整个网站一定会出现各种各样的故障，关键是在故障出现以后，应对能力有多强，恢复

时间可以做到多短。追源会在QCon上分享，我们在应对不同类型的故障时，我们是怎样去恢复的，恢复时间能控制到多短，为什么能控制到那么短。在不同的技术能力，以及不同的技术设施的情况下，能做到的恢复时间可能是不一样的，所以我们会定义一个在不同的技术能力和不同的技术设施的情况下，恢复时间的标准。如果恢复时间能控制得非常好，可能整个故障控制力就非常强。举个例子，比如淘宝因为能够做到异地多活，并且流量是可以随时切换的，所以对于我们来讲，如果一地出现故障，不管是什么原因，最容易的解决方案，就是把这一地的流量全部切走。这样可以把故障控制在一分钟以内，整个可用性是非常高的。

关于系统的容灾能力，国家也有一个标准，最重要的一点就是故障的恢复时间。如果大家都以故障恢复时间控制到哪个级别来衡量，那所有网站就有了一套标准。

InfoQ：好的，异地多活我们就聊到这里。您现在在维护一个叫做HelloJava的微信公共帐号，您在工作中还是会去调优一些具体的Java问题吗？

毕玄：没错。我在阿里巴巴这些年来，很多问题有一些会流转到我这里，或者有人会反馈到我这里，所以我会介入到很多问题的排查，现在也是一样，有些问题我可能就会介入，直接去排查。为什么有HelloJava那个微信公共账号呢？最大的原因也是因为，我们都知道很多人会排查故障，有些人可能只是一眼扫过去就已经知道原因是什么。为什么呢？他可能已经排查过很多的故障，积累出了经验，并不一定是这个人的能力比你强，可能就是因为他见的比你多，他对工具使用的熟悉程度比你强。比如说我们看到故障很多，排查故障很厉害的人，他可能就是会善于用各种各样的工具，而且用的非常熟练。所以我做自己的HelloJava，就是希望有更多的人看到，我们在面对一个故障的时候是怎样处理的。希望大家即使没有处理过这个故障，至少也看过一篇文章，也许在下次面对这个故障时，至少有点思路，知道应该怎么去处理。这个也是阿里巴巴分享给外

面的一些经验，很多时候就是经验的积累。

InfoQ：您最近在HelloJava里提到了一些期待的Java特性，您这边会将它们反馈给Oracle，让他们加入吗？

毕玄：我们跟Oracle官方有过一些交流，谈到我们期望的JVM的一些改进。但是说实话，因为我们看到的很多JVM的问题，可能是因为我们流量比较大，我们对Java的性能、高并发有很高的期望。如果能够突破一点，对于我们整个网站来讲，提升可能就是巨大的。但是对于Oracle来讲，因为OpenJDK不仅仅是为大公司做的，而是为所有不同的用户，比如中小企业、大用户，所以他们必须衡量需求的分布。所以有可能我们的需求，对于Oracle的官方JVM团队而言，只是小众需求，他们不大会投入很大精力去做。

InfoQ：所以您这边的团队会做一些定制？

毕玄：对。很多人知道，赵海平加入了我们的团队，如果知道他背景的话，知道他以前做过PHP运行引擎的开放，不仅仅是HipHop的翻译，还包括怎么运行整个PHP应用程序。其实你可以认为也会类似VM，因为我们已经看到在Java、JVM的哪些点上如果有突破，可以给我们带来巨大提升，所以既然官方很难往前推进，那我们就自己来推进。

受访者介绍

林昊（花名毕玄）于06年加入淘宝，为阿里巴巴集团核心系统资深技术专家。目前是阿里巴巴技术保障部的研究员，负责性能容量架构。

京东云首席架构师刘海锋： 私有云建设的挑战与应对之道

去年的双十一过后，InfoQ 曾经采访过京东云平台首席架构师刘海锋。

3月31日，在华为ICT巡展北京站的活动中，刘海锋分享了《京东基础云服务技术演进》。

基础云服务支撑着京东很多业务的发展。它可以分为三个层次，包括底层的存储服务，核心的中间件以及上层的弹性计算云，通过 API 以服务的形式支撑其他业务单元。下面我们分别来看一下。



存储系统面对的挑战与应对之道

存储是互联网公司最基础的东西。这也是开发团队花精力最多，持续迭代的一个技术方向。

挑战 1：非结构化存储

京东每天有千万级的商家上传图片，用户浏览完图片后会产生交易订单，需要很多文本来描述这些订单。另外京东有自己的库房，任何一份订单经过拆分，经过库房的流转，每一分订单又会产生几十份非结构化的数据，涉及商品的入库、出库、调拨等。商品送到客户手中之后，还有签单信息、银行小票，未来这些信息的电子化、和银行对账，又是很多非结构化的数据。

- A. 商品图片：千万/天
- B. 交易订单：千万/天
- C. 库房记录：亿/天
- D. 电子签收：千万/天

非结构化的数据越来越多，如何存储这些数据？商家的图片、交易的订单、库房的记录、电子签收的信息，这些数据都是非常关键的，而且这些数据有一个特点，量比较大，但每份数据一般都比较小。

JFS:Jingdong Filesystem

京东针对非结构化数据开发了大规模分布式存储系统JFS（Jingdong Filesystem），支持BLOBs/files/blocks。现在这个系统已经到了3.0版本，可以统一管理小的对象、大的文件以及私有云中可持久化的块设备。

技术方面，因为数据不能丢失，所以从一开始就讲究强一致的复制，使用了Paxos复制；在存储引擎以及各种数据模型上采用了统一的存储管理；随着规模的增大，到了几PB数据的时候，采用了Reed-solomon码来降低存储成本；元数据的管理和Hadoop的集成方面，目前也有不错的进展和落地的应用。

目前该系统已经在支撑京东商城的如下服务：

- 图片服务
- 订单履约
- 物流数据交换
- 电子签收
- 内部云存储服务
- 虚拟机与容器卷存储

挑战2：越来越多的缓存

为保证快速响应，很多数据都会放到内存里，比如商品的价格，搜索推荐的结果等。越来越多的缓存，越来越多的大内存机器，不同的业务，如何管理它们也是很大的挑战。

最早，很多小规模的公司可能会采用 Memcached、Redis 等，但当到了很大规模的时候，技术也会发生质变。

Jimdb：分布式缓存与高速 NoSQL

Jimdb 是京东研发的企业级 NoSQL 服务，能够统一做分布式的缓存，也能做高速的键值存储，完全兼容 Redis 的协议。与 Redis 相比，它有如下特性：

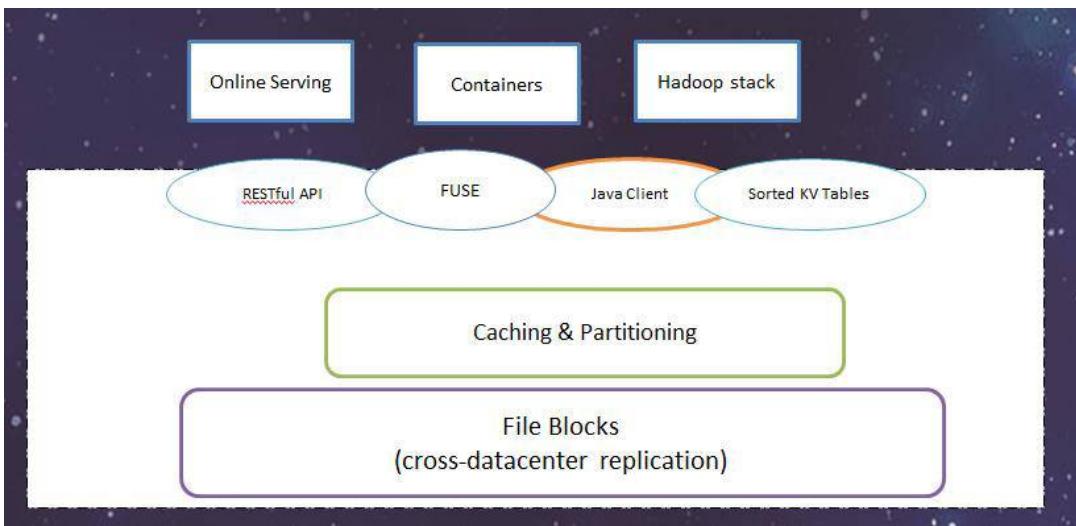
- 精确故障检测与自动切换
- RAM/SSD 混合存储
- 在线横向扩容
- 异步、同步、局部复制
- 全自动化接入与管理

其中“全自动化接入与管理”这一点是最近半年的主要工作，目的是降低维护成本。

Jimdb 在京东部署了 3000 多台机器，都是大内存+固态硬盘的，支撑了京东的商品详情页、搜索、推荐、广告点击等很核心数据的快速访问。

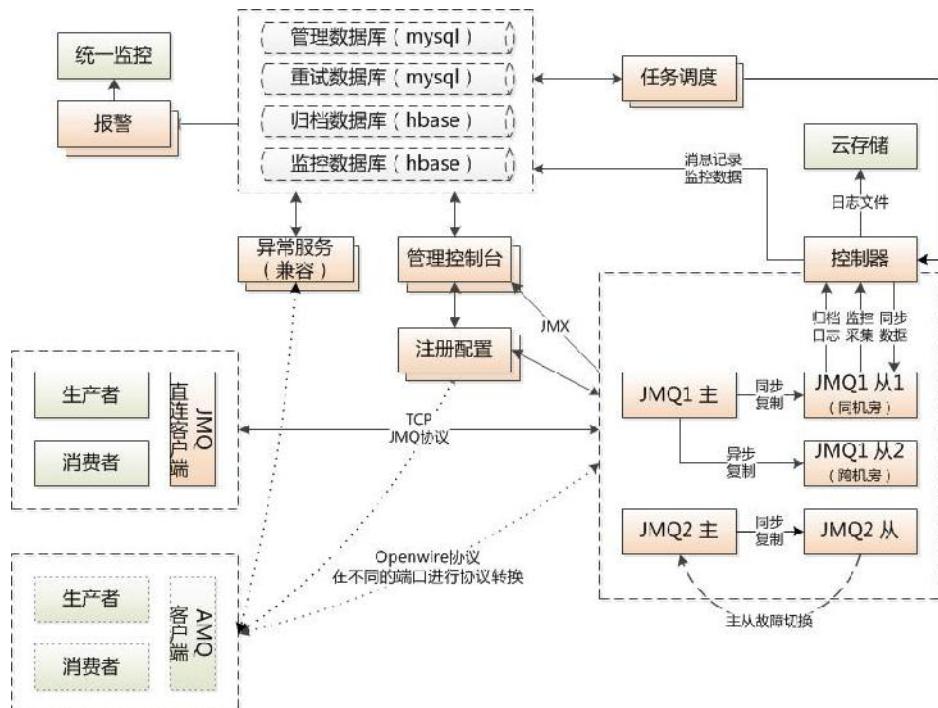
下一代新存储平台

刘海锋的团队最近在做的事情是，更多考虑多数据中心的复制，让数据更加可靠，并且希望做统一的存储服务，实现 One Jingdong One Storage，用一个系统抽象出文件、对象、表甚至缓存，能够在分布式的多个数据中心实现统一的复制存储底层的数据，在主 IDC 做缓存，甚至全量内存加速；向上支持在线服务、Hadoop，支持私有云内部的容器卷的管理等。



中间件：消息队列与SOA

底层存储的上面就是各种中间件。



挑战：越来越多的消息传递

京东跟很多互联网公司不一样的地方是，除了北京和江苏的核心机房，在全国各地还有很多商品的库房，每个库房会有几十台服务器，相当于一个小型的数据中心，消息队列不仅要串联核心机房里的业务，还要驱动库房里的订单生产环节，跟业务存在很强的依赖关系。从订单管道，到核心机房，再到仓储库房，普遍是用消息队列驱动业务流程的。日均消息数超过百亿。

JMQ : Jingdong Message Queues

面对业务挑战，京东的消息队列系统经过了三代演进，于去年双十一之前上线了 JMQ 并完成切换。

JMQ 有如下特点：

- 机房断电不丢消息。因为库房的网络环境是不稳定的，所以必须保证不丢消息。
- 组提交技术。引入了数据库中经典的 group commit 技术，提高同步刷磁盘的写性能。
- 透明压缩
- 灵活复制

挑战：越来越多的在线服务

电商系统内有很多服务。这些服务内部会互相调用，对外也要开放一些接口，供商家或者合作伙伴使用。

JSF : Jingdong Service Framework

京东在这方面的解决方案是JSF。可以做到运行时服务质量分析，提供了完善的服务治理功能。目前在京东已经接入了几万台服务器，更好地支持了内部的SOA化以及对外的服务开放。

弹性计算

弹性计算云是目前的主要工作。

挑战：越来越多的机器

任何一个高速发展的互联网公司，机器数量都是在成倍增加的。随着业务规模的增长，机器的数量也在不断增加。现在就要面对这样的场景：有很多数据中心，而每个数据中心内的机器又会被划分给不同的业务，比如有的机器处理交易，有的处理订单履约，有的处理搜索，有的处理图片等等，面对这么多机器，应该怎样管理，让研发团队的效率更高呢？

弹性计算云

该项目的愿景是在IDC的资源和业务系统之间建立一个桥梁，让业务与机器完全解耦，做到真正自动化维护，缩短产品开发到上线的流程，让工程师的精力更多放在产品的设计和研发上，而不必关注如何申请资源、如何上线；提高资源利用率和服务质量，让研发团队的生活更美好。

从公司整体来看，希望能够提高资源的利用率。因为很多机器是分散的，由各个业

务团队使用，必然有很多机器是空闲的，统一管理必然能提高资源的利用率。

弹性计算云的整体架构如下图所示，分拆两层服务。底层是基础服务，实现软件定义数据中心。通过OpenStack和Docker的结合，实现容器化。JFS实现可靠的存储。上层是平台服务，希望通过集成部署，实现资源的统一分配，业务不用关注自己到底部署到哪台机器上，并由平台根据业务量实现自动伸缩。



现在这个系统已经在部分业务中落地，大规模落地会在今年年完成。具体而言，像商品详情页、图片系统，就是弹性计算云支撑的，用户的每一次浏览，都有这个系统在做贡献，能够按访问流量自动调度资源。在流量高峰的时候，这两个服务会自动扩容。

做为总结，刘海锋总结了两点：

1. 业务发展推动基础架构的演进；
2. 技术的关键是团队。

在演讲之后，InfoQ采访了刘海锋。

InfoQ：最近两年，容器技术，特别是Docker在业界非常火，可以介绍一下京东在这方面的实践经验吗？

刘海锋：刚才我在演讲中提到，我们希望在IDC的资源和业务系统之间建立一个桥梁，让业务与机器完全解耦，这是一个很复杂的工程，不是一个容器或者Docker就能解决的。弹性计算云分为两个层面，底层的基础服务更多是把机器做一个统一的虚拟化、容器化，上层的平台服务更多地要考虑怎么样去配合京东的业务，跟应用能够更顺畅地融合在一起。

举个例子，从最简单的部署角度来说。以前部署是这样的，一打好包了，在部署界面上选，选了三个机器，三个机器的IP分别是多少，然后部署。部署完成检查一遍就成功了。而接下来整个系统会做成这样，一个程序要上线了，要部署，点部署，系统给部署完成之后告诉开发者成功了，然后再说部署在哪里。不需要关心之前的那些环节，不需要层层的领导审批和运营部门审批。而是直接部署，部署位置由平台来统一控制。一开始分配很少的机器，随着流量的增大自动扩容。需要的量小的话，再自动缩容，节省出的机器再统一调度。这样可以提高公司的资源利用率，数据中心中都是公司的机器，而不再分你的我的。

我们用Docker，更多的是考虑它比较适合公司内部的私有云环境，而且比较轻量。我们底层的平台，简单的理解是Docker和OpenStack的嫁接。我们用OpenStack去管理Docker，比如给它分配一个独立的IP等。但是除了两个第三方平台，我们的底层平台要做到简单、可控、稳定，所以上面还有一个很强大的完全自研的一套平台和服务，能够统一的调度和控制，实现管理、监控和部署。

Docker在这里面发挥了很重要的作用。另外我们也稍微做了一些改造，去掉了很多用不到的特性，让它稳定简单。

网络方面，用Open vSwitch给每个Docker容器分配一个IP，这样每个容器看上去跟虚拟机或者物理机没什么区别，迁移业务的时候大家更容易接受，过渡更为平滑。

InfoQ：刚才您提到了OpenStack和Docker的嫁接，可以具体介绍一下其结构吗？

刘海锋：基础服务层面，核心有三点：OpenStack、Docker和自研的JFS存储。OpenStack和Docker在这里都是非常重要的组件。

先说OpenStack。因为OpenStack变化比较快，我们没有那么多精力一直跟随。我们从OpenStack拆出了一个分支，经过定制，做了一个内部的分支，叫做Jingdong Data Center OS。这方面的工作有两点。一方面是让它能跟Docker更好地配合；另一方面，我们加入了很多故障处理功能，应对物理机故障和容器故障。物理机故障的时候快速检测出来，快速报警，快速迁移上面的容器。

再说Docker。网络和存储上都有些改造，比如可以支持JFS mount过来的目录。我们会把故障当成常态，物理机的硬件故障，像磁盘、内存和硬件方面的问题，所以对OpenStack做的改造主要是更好地应对故障，快速响应。

InfoQ：Docker或者OpenStack方面的工作，会向开源社区反馈吗？

刘海锋：其实我们也有考虑。我们计划在今年年底或者明年初的时候有些动作，目前还是希望把当前的工作做好。

InfoQ：社区的版本更新比较快，这方面如何merge自己的特性和新版本的特性呢？

刘海锋：外面比较好的我们也会吸收。实际上OpenStack更多是面向公有云设计的，从我们的需求来看，它比较臃肿，不是很好控制。所以我们做了裁剪，砍掉了很多公有云的功能。

InfoQ：这个会不会像你们的文件系统那样，在定制的过程中走向自研了？

刘海锋：这是有可能的。像 Docker 这样的项目，设计的时候考虑的是更为通用的目标，功能非常多，这会引入过多的复杂性，也引入一些问题，所以要做减法。而且就互联网公司的开发团队而言，基础架构是在业务的推动下不断演进的，而不是满足某种通用的需求。

InfoQ：现在弹性计算云的落地情况如何呢？

刘海锋：现在弹性计算云已经在支撑很多业务，像图片服务。很多边缘系统也在用。根据研发战略，我们希望今年大规模落地。

InfoQ：可以介绍一下公司其他业务在向私有云迁移的过程中都有哪些挑战吗？

刘海锋：挑战很多，我们很多精力也放在这上面。我们做了一个基础设施，会面对两类用户，像新的用户和新的产品，直接选择它就可以了；但像老的服务，可能占用的是物理机，资源利用率很低。迁移其实不仅是技术层面的问题。所以我们会有专门的项目经理团队去推动。另外我们有一些工具，方便旧业务的迁移。

此外，我们还需要在保证业务稳定的前提下追求规模，因为规模大了才有资源管理方面的优势，所以我们希望做到万台规模。我们每天都很谨慎，很小心，但是还是希望量上来。

在 4 月 23 日~25 日的 QCon 北京 2015 大会上，刘海锋将担任微服务架构专题的出品人。在该专题中，京东云平台高级架构师、JFS 项目的负责人李鑫将分享京东的服务

化框架实战。

受访者介绍

刘海锋，2013 年加入京东，担任云平台首席架构师、系统技术部负责人，主持建设存储、中间件、弹性计算等私有云技术体系。



华三通信研发副总裁王飓：传统通信技术与云计算的关系

云计算无疑是IT界近几年最火的话题之一。很多互联网公司，传统IT公司，甚至是初创公司，都将视角投到了云计算上。

在于2014年10月16日-18日举行的QCon上海技术大会上，华三通信研发副总裁王飓做了题为《SDN控制器集群中的分布式技术实践》的主题演讲。在大会召开前，InfoQ中文站就传统通信技术与云计算的关系对他进行了采访。

InfoQ：在云计算时代，通信技术历史的积累您感觉有多少是有用的？新的技术又是主要来自哪里？

王飓：通信技术历史的积累其实大部分都是有用的。云计算核心关注的是虚拟化，并没有破坏TCP/IP的通讯层次模型，也没有彻底改变网络上需要的各种网络设备的种类（比如依然需要交换机/路由器/防火墙）。只是这些设备的控制管理方式（SDN是改为了集中式管控）和形态（产生了虚拟形态如VNF）发生了变化。当然个别协议在新的SDN架构上可能没用了，或者需要进化，这就是很自然的事情了，任何协议都有生命周期。

InfoQ：基于已有的体系改造，要比新建设一套体系更加困难。无论如何，过去有很多积累要放弃，给新的技术腾地儿。您目前跟通信领域

的同行们沟通，感觉行内整体目前的心态是偏激进的（主动引入新技术革自己的命），还是偏保守的（用各种手段拦着新技术发展，给原来的积累多争取一些时间）？偏激进的那批人主要在哪儿？

王飓：我感觉业内大部分人的心态是比较积极的。现在谈革命可能还不够准确，就像我上面说的，云计算时代，并不是完全颠覆已有的东西，那些过去的积累并不会一下子失去价值。

其实，大概在6、7年以前，我也迷茫过：数据通讯网络发展到当前的状况，是否就差不多了，剩下的只是提高一下带宽，提高一下芯片的容量就可以了？这就好像19世纪末一些物理学家认为经典物理学大厦已经建成，后人只有拾遗补阙的份了。所幸，这个世界的精彩总是超过我们的想象。云计算/SDN等技术的发展，无疑为我们的未来又重新增添了无穷的魅力。

当然，新技术是否就一定是好的？这个有待时间来验证。即使方向是对的，但具体的技术路线可能还是需要不断探索。所以，如果单纯是技术手段，则不存在什么保守或者激进的说法，任何技术行为都可以看成是一种尝试。在通信产业界，各种通信协议的竞争一直都有，而胜利从来不是单纯的看技术先进与否，适合的才是最好的。（当然，如果利用自己的垄断地位采用非正常的竞争手段是例外。）

我和我们公司都是这种变革的积极拥护者，这种变化，无论如何，最终都是让我们的技术更贴近我们的用户，让网络世界变的更美好。

InfoQ：有观点认为互联网是IT的消费者，并不是主要的IT贡献者，而主要的IT贡献者其实还是IBM、思科、华为这些传统IT公司。现在IT越来越便宜，互联网公司获益最大，IT公司、通信公司则很惨。但其实我们也看到像是Google、Amazon这样的互联网公司也在越来越多的往基础IT做投入。您如何评估现在互联网公司对于整体IT的贡献度？

王飓：如果IT这个词代表网络基础设施的话，可以认为互联网公司是IT的消费者，但别忘了最终的消费者是广大的用户，所有的评价应该以最终用户得到了什么作为评价。

技术本身是没有边界的，无论是互联网公司，还是所谓的传统IT公司，最终的目标都是满足和提升广大用户的使用体验。如果互联网公司认为现在的网络基础设施阻碍了他们提升用户体验，现有的传统IT公司无法给出他们满意的解决方案，则他们在这个方向投入就是很自然的事情。

至于说IT公司和通信公司，很惨应该还不准确，应该说是其利用技术和市场垄断来谋取高利润的时代结束了。

我们或许可以这样理解，互联网公司就是第三产业，是服务业，IT公司就是第二产业，是制造业，完成了工业革命以后，第三产业的比重越来越大，附加值高，这是历史大趋势。这也是为什么十多年前思科、微软这样的公司市值最高，而现在的新兴是Google、Facebook和阿里巴巴。而当第三产业高度发展以后，肯定会反哺第二产业，这也就是为什么互联网公司开始向基础IT投入的原因。

想精确评价互联网公司对于整体IT的贡献度恐怕是很难的，但毫无疑问，他们的影响是正面的，推动了IT的发展，最终用户得到了个更优质、更方便、更便宜的服务。

InfoQ：互联网公司作为用户，在网络方面倾向于OpenFlow这种完全由Controller掌控的网络结构，比如Google和Facebook都是OpenFlow的主要推手。但是这似乎是早期的一个观点，发展到现在，他们理想中的Controller还是没有实现，而且现在越来越多的声音也认为将所有控制逻辑放在Controller里面也未必合理。您对此是什么观点？您认为哪些节点应该使用Controller，而转发层面应该保留哪些逻辑？

王飓：说这个问题，先看看我们的世界：今天的人类社会，是由一个个国家组成

的，所有国家构成了一个松散的组织——联合国。每个国家都是一个自治的区域，有各种各样的政体，有的是松散的联盟，有的中央集权。看看互联网发展史，是否也是这样呢？我一直认为，网络虚拟空间，就是现实的折射。

所以，在一些封闭的空间里，比如一个数据中心，一个小的运营商网络，是比较适合使用Controller集中控制的。而在一些开放空间，边界不是很清晰的地方，还是需要传统的网络，这也是互联网诞生之初的设计思想：在一个无比广阔的空间里，网络自由互联、互通、自治，没有一个集中的控制点，任何对网络的攻击都只能影响局部，不会导致整个网络的崩溃。

至于转发层面需要保留哪些逻辑，现在大家争论还是很热烈的，但显然完全没有逻辑已经被证明不是最理想的，因为今天的设备，即使是冰箱和洗衣机都可以变的很聪明。这样看，完全把转发层面当成无逻辑的节点显然也是一种浪费。目前看，一些需要高速检测、本地链路快速切换等可能放到转发节点上更合适。

InfoQ：现在互联网公司做云计算，传统IT公司也做云计算，运营商和通信公司也做云计算，初创企业也做云计算——这里说的云计算仅限于IT基础设施层面。就您的观察，你觉得他们要做的东西有什么不同？想要达到的目的有什么不同？对技术的需求又有什么不同？

王飓：云计算这个概念很大，大到不同的人可以有不同的理解。云计算的世界很广阔，广阔到可以容纳这些不同的力量一起来建设。

从方案上看，大体可以分为公有云，私有云和混合云三种；从内容上看有的偏重计算，有的偏重网络。

从技术本质上看，他们做的东西是类似的，但从需求角度看，又有明显的区别，各自的侧重点也不同。

互联网公司和运营商本身都是提供服务的，都是要解决自己的网络所面临的问题，

相当于私有云，然后是建立公有云，提供公有云服务。

而通信公司则是希望借此机会进行转型，由卖设备变成卖服务，因为大家看到了，随着这个趋势（硬件标准化/软件开源化），设备的毛利率越来越低，价值链中服务的部分会逐渐成为主体。这些通信公司即做公有云，也做私有云，甚至是混合云。这个服务和前面讲的互联网公司的服务不同，是指提供完整解决方案服务，自己并不运营。当然也不排除某些企业借此转型，变成了互联网公司。

初创企业就不说了，这个领域肯定是拿VC的热门，为什么不搏一下？而且初创企业没有历史包袱，显然在技术上是最有冲劲的，因为传统领域的蛋糕已经分完了，后来者要么重新做一个大蛋糕，要么就是打破原来壁垒，而推动一次技术革命显然是一个不错的选择。

InfoQ：感谢您接受我们的采访。

受访者介绍

王飓，华三研发副总裁。从事数据通讯设备软件开发长达14年，作为资深的网络协议专家和软件系统架构师，熟悉多个层面的数据通讯协议，擅长做通信协议设计以及实现，对嵌入式系统和复杂软件系统设计，以及对实时系统的性能优化有着十分丰富的经验。此外，对网络安全有着比较深入的研究，对各种网络攻击和防护有着丰富的经验。近年来开始关注并投入SDN相关领域的研究和开发。对OpenStack、OpenDaylight、OpenVswitch、NFV等都有一定的研究，对云计算时代的网络通信有着深刻的理解。

明略数据CTO冯是聪：打造最易用的跨平台数据整合系统

明略数据是一家聚集了国内顶尖大数据人才的技术型大数据整体解决方案供应商，其从创立之初就秉承着将技术研究落地转化为科技生产力的基本理念，至今已经为银联、中央电视台、中国联通、国美在线、苏宁云商等公司部署了大数据处理平台，并带来了大量的业务创新机会。那么，明略数据是怎样做到这些的？明略数据在技术层面上又具有怎样的过人之处呢？为此，我们请到了明略数据CTO冯是聪博士进行了采访，以便更加深入的了解明略数据的技术特点。

InfoQ：明略推出的大数据平台BDP，对于这个平台我理解的就是很多传统企业比如说银行、政府，这种大型的机构当中，会有很多的分支部门，而部门之间的数据可能会由于种种的历史原因无法进行打通。这些数据，可能它的字段跟描述方式以及存储的格式也是不一样的。那么该如何把这些不同格式、不同表达方式的数据进行打通？是不是BDP这个产品可以实现这样的功能呢？

冯是聪：从技术上讲，对于一些企业、政府机构来说，一定会存在这样的情况，它有不同的数据来源的，不同的数据格式。那么这些数据必然面临着一个问题，就是如何把它们融合在一起，怎么实现数据之间的交互。

这一问题从技术的角度上来看确实具有一定挑战，但明略恰恰就善于解决这种问

题。明略BDP中有两个核心模块——Data ONE与SQL ONE。Data ONE采用的是All-In-One模式，无论数据来源是什么，无论是来源于关系型数据，还是来源于非关系型数据库，是NoSQL，还是来源于NewSQL，或是文件系统，这都没有关系。明略会以统一的方式将这些数据放到BDP平台内，通过Data ONE把所有数据统一管理起来。

那么接下来怎么实现数据之间的交互呢？这就需要用到另一个核心模块SQL ONE了。SQL ONE是一个标准的SQL查询引擎。传统的新客户一般对于关系型数据库都非常熟悉，对SQL语句也会非常熟悉。那么当我们提供了SQL ONE这种语言之后，如果客户会操作传统的关系型数据库的话，就可以操作我们所有的文件系统、NoSQL，甚至是NewSQL。SQL ONE可以智能地识别这些数据被物理地存放在Data ONE的那个子系统中，确定数据是放在关系型数据库，还是放在非关系型数据库，或是放在文件系统中。客户只需要输入一个SQL语句，系统就能自动完成所有的事情，这也是BDP的一个特点之一。

InfoQ：从数据安全问题上来说，不同的行业，不同的企业，对数据安全的审计、审核的标准也不一样，尤其像一些涉及到国计民生的政府机构，他们的数据对安全的要求是非常高的。明略的产品是部署在客户的数据中心当中的，这样从物理上就可以规避一部分安全隐患。那么除此之外，明略还有在安全方面还有哪些不一样的地方？

冯是聪：从目前来讲，在大数据安全这一领域中很多技术都是不太成熟的。从大数据的特点来看，首先数据规模比较庞大，数据内容也比较复杂，再加上各种数据来源，各种数据格式，还要要求统一在大数据平台上进行管理，这些因素导致其对安全技术的要求变得非常高。

明略针对这些问题开发了自己的核心安全组件Acre，在Hadoop平台上首次实现了行列级别的数据安全访问管理。它的核心思想是，可以把任何人操作该数据的历史、权

限，包括他的授权认证，全部统一管理起来。

另外在隐私保护方面，明略实现了多种数据脱敏与加密算法，智能地实现了敏感数据的自动脱敏和保护。

InfoQ：您刚才也提到，明略还会在数据价值挖掘上有一些自己的动作，这就可能涉及到机器学习、深度学习，这些现在比较流行的新技术。那么，能否介绍一下明略在这方面的一些研究实践？

冯是聪：机器学习还有数据挖掘是大数据最核心的技术之一。明略的3大核心产品之一的DataInsight就是数据挖掘和机器学习的一个典型的平台。数据挖掘和机器学习在明略实施的几乎每一个项目中都得到了充分地应用，基本上每个项目都会进行一些预测、分类，这些都会用到机器学习里面去，另外像以前机器学习有进度学习、无进度学习、深度学习，这些也都会用到明略的项目里面去。

InfoQ：展望2015年，您认为哪些类型的企业会成为大数据领域的明星企业，或者说哪些企业会有高速的增长空间？能根据您的研究，分享一下您的观点吗？

冯是聪：因为大数据现在已经慢慢被大部分企业或者是政府接受了，它会在很多的领域都得到广泛的应用。从我个人看来，我觉得有两个领域是值得关注的，第一个是金融领域。现在的个人贷、余额宝等金融产品越来越多，因此为了更有效的进行反欺诈，征信系统将会利用更加密切的、彻底的应用大数据技术。

第二个领域是安全领域。安全永远都是一个话题，几乎每一家企业、每一个政府机构都会关心安全问题。数据安全技术没有得到突破的情况下，很多企业和政府是不会轻易的把自己的数据放在云端的。另外现在有的公安机关，甚至军方机构，都开始将大数

据安全技术用于追捕或是反恐，这都说明了安全领域将更多的应用大数据技术。

InfoQ：明略的商业模式是很清晰。那么在未来，您更看好是像明略这样的面向企业的 On-Premise 的商业模式，还是同时还看好别的一些大数据创业公司的商业模式？

冯是聪：对于我自己来讲，我肯定是看好明略的商业模式的。一方面这种模式能够更好的基于客户的不同需求进行定制化开发，另一方面在安全上也更有保障。那些能够跟客户共同成长，能把客户当成伙伴，能够把客户的问题当成自己的问题的那种公司，才能够得到比较迅猛的发展。

大数据的核心在于从数据中挖掘价值。2015 年是大数据应用元年，企业将更加关注大数据技术的落地和应用。因此我比较看好那些能够根植于客户业务，能够帮助客户解决业务痛点，真正能够给客户带来价值的大数据公司。那些在不同细分领域，能够提供整体解决方案的大数据公司的前景将更好。

InfoQ：也就是不仅仅要做技术，而且还要熟悉、了解客户的业务模式，从而能更好提供有针对性的大数据服务。

冯是聪：明略始终认为大数据仅仅靠技术是不行的，它必须要能解决业务问题。厂商的数据科学家通常需要三方面的知识，一方面是需要懂得计算机知识，第二方面他要懂得数据挖掘知识，第三方面他要懂得数学，这是综合能力的体现。而只有当把客户的业务本质了解比较透彻，才能给客户带来实际的价值。

InfoQ：您能否谈谈有哪些技术会对大数据行业的未来产生巨大影响或者说带来巨大推动力？

冯是聪：我认为有四类技术比较重要。第一类技术是大数据安全技术，无论是金融行业的反欺诈，还是警方的反恐与安保，都需要有大数据安全技术的帮助。

第二类技术是机器学习领域，从各种报道来看，无论是在云识别，还是图像识别，甚至视频的处理，已经基于机器学习以及深度学习而得到广泛的应用，我相信随着深度学习的发展，将会带来巨大的变革。

第三类技术是量子通讯，据我了解中国量子通讯的研究还是非常的具前沿的，基本上处于国际领先地位。像中国科大，他们现在在量子通讯上，能够在超过一百公里上午距离上进行传输。所以我相信随着量子通讯技术和量子计算机的发展，最后我们的通讯技术，还有计算机技术、语言都会发生翻天覆地的变化。

第四类是智能设备。我们身边生活中的几乎每一样设备，每一样东西实际上都可能会智能化。而一旦设备智能化了，这就需要想办法将数据收回来，当这些数据达到一定规模的时候，就一定会需要大数据技术来进行处理这些数据。我相信随着智能设备的发展，无论是中国还是外国，人们的生活方式以及工作方式都将得到改革。

受访者介绍

冯是聪，北京大学计算机系博士毕业，北京明略软件系统有限公司联合创始人兼CTO

UnitedStack 创始人程辉：互联网精神 + 开源战略 = 成功的托管云

本次受访嘉宾是 UnitedStack 创始人程辉，就云计算市场的现状、发展趋势，以及 UnitedStack 在业务方面的战略调整给出了自己的解读。

InfoQ：为什么 UOS1.0 是做发行版，而从 2.0 开始做公有云和托管云了？

程辉：公司 2013 年成立，在当年 10 月份的时候发布 UOS1.0，当时的想法很简单，很多厂商都推出高度产品化、定制化或者优化过的 OpenStack 发行版，然后通过外围的一些服务挣钱。我们也想解决 OpenStack 的一些痛点，比如自动化部署、运维等，并针对国内用户的使用习惯进行了改进，最终发布了 UOS1.0。产品本身是比较酷的，把 U 盘做成了一个产品，交付给任何一家 IT 公司或者个人用户，在服务器上插上 U 盘，过一会就搭建出一个云环境。

但我一直在反思。用户拿到了 UOS1.0 之后，整个安装过程非常快捷，但是用户拿 UOS 1.0 来提供 7x24 小时持续的云服务还是很遥远。我们只是解决了从无到有的问题，而这只是万里长征第一步，接下来还需要提供对外服务，保证产品不宕机可扩展，而当时我们并没有解决这个问题。

所以，公司做了重大的业务转型。把 UOS 1.0 中的核心技术包括分布式存储、高性能网络、优化的主机调度等，应用到自己的公有云上，开放给公众使用。当时还没有

考虑商业模式的事情，只是觉得我们应当把这些有价值的技术和产品开放出去，让别人受益，公司就自然就有价值了。说做就做，我们拿出了公司剩余的大部分钱在北京租了机房，买了一批设备，从核心技术到计费平台、说明文档、注册系统、自动化运维等，花了近半年的时候做公有云。

InfoQ：公有云发布之后遇到了哪些挑战？

程辉：主要有三个挑战：

第一，如何在坚持OpenStack开放标准的同时满足国内客户定制化的需求。UnitedStack云服务完全基于OpenStack开放API构建，但是OpenStack开放API并不能完全满足客户需求，因此这里需要与社区做足够的沟通工作，将这些差异化的需求提交给社区，同时我们还在保证100%兼容的目标的情况下对OpenStack API进行扩展。这对于团队对于OpenStack开发能力有足够的自信才能做到。

第二，平衡OpenStack社区开发与生产运营的差异。社区开发时，我们只需要完成功能开发和测试，但当我们要生产运营一个OpenStack云平台时，这时需要考虑平台运营过程中可能出现的各种事件，比如物理服务器宕机，存储扩容、缩容，磁盘故障，网络抖动和攻击等，需要为每一种异常或者失效准备预案，及自动化运维措施，并及时响应。

第三，获得客户信任。作为一个新兴公有云平台，获得客户信任是一个漫长的过程，任何一次异常或者故障都会导致客户信心的丢失，客户几乎不能容忍一次故障，这是最大的挑战。平台每天都会有更新和升级，也不能中断客户业务。

InfoQ：UnitedStack为什么提供托管云业务，出于什么考虑？

程辉：有句话说“出来混总是要还的”，刚开始创业的时候，我们没想商业模式，从发行版到公有云，都没想好怎么赚钱。我们知道现在很多公有云都是巨头在做，几十

亿的资本投进去才可以做好。作为一个小的创业公司做公有云，你确实有机会，但是相比资本的力量，这是上百倍的差距，你在市场上可能有竞争力，但是很难做的比他们更好。

我开始思考如何进一步商品化整个公司的品牌和技术，在国内，有一批大客户，对云的需求量更大，而且没有哪一家公有云可以服务好他们。大到什么程度呢？大到用公有云已经很不划算了。比如对弹性计算要求极高的新兴的移动互联网公司、游戏公司，还有对云扩展性和安全性要求高的银行和互联网金融公司等，他们的业务量规模大且比较需求量比较固定，而且对于安全性、数据主权等要求极高，因此这些客户不太放心将这些业务放到公有云上。

所以，我们推出了托管私有云(Managed Private Cloud)，可以理解成独享的公有云。我们的核心价值在哪里？我经常把云建设的投入分为三个部分，一是IDC资源，包括电力、带宽、机位等，这是一个高度市场化的领域，比较成熟，这块交给客户去解决，因为价格已经市场化了；二是服务器设备，更加市场化的领域，发展了几十年，我们没有必要做；三是独立的技术平台和运维，这才是我们应该做的事情，帮客户做好管理、维护以及后续的升级，甚至新功能的研发、监控等。

事实上，如果把托管云三部分的投入成本和同样资源的公有云费用做比较，就会发现，托管云的整体成本只有公有云的 $1/3\text{--}1/5$ ，看起来不可思议，但事实如此。目前，已经有 10 个托管云的大客户上线，机房 12 个，分布在北京、广东、上海和东北地区。

我可以随口算一下，做一个云计算环境，需要的人包括虚拟化工程师、存储工程师、网络工程师、监控工程师、UI 设计师、运维工程师等等，每一个岗位都需要花很大价钱。托管云可以让客户节省大量的钱，关注自己的业务。在 UnitedStack 平台，托管云的系统平台和公有云是一样的，有什么更新，都会同步升级。

InfoQ：既然托管云商业模式比较好，为什么还要做公有云，据我所知国内的其他公有云市场盈利艰难。

程辉：这是个好问题，很多人都不理解。在没有公有云之前，我们去向客户推销技术平台时，客户经常会觉得你说的这个好东西没有经过验证，没有看到实际的生产案例，没有看到实际的用户，后来，我们上线了公有云，让大家看到我们的高性能、用户体验、运维、持续更新等能力，通过这些方式，客户才开始接受我们的托管云。另外，不同企业，在不同的阶段，对云的需求是不一样的，比如，互联网创业公司，肯定初期倾向于公有云，待业务规模足够大而且稳定的时候，这时采用第三方服务的私有云可能是一个更好的解决方案，他们需要不同的云服务模式去支撑他们当前的业务。因此，总结一下，公有云一方面满足部分客户的需求，另一方面，方便客户构建其混合云体系。因此，这里公有云也是我们商业模式的一部分。

InfoQ：关于托管云服务，用户自己找机房和数据中心，那么在搭建和维护云服务过程中，是不是偶尔需要你们派工程师去现场？

程辉：我们现在落地了10个大规模的托管云，几乎没有上门服务过！前期，我们会和客户商量好，需要采购哪些设备，如果配置，发给他们一个表单，购买之后，我们的工程师会告诉他们如何关联这些设备，还是一个清单搞定。最后是打通VPN隧道，一旦完成，我们就可以通过远程方式部署第一台种子机器，剩下的其他机器就会逐渐配置完毕。我们最快的客户案例是从确定合同到托管云正式上线用了不到一个月的时间。我认为，以云计算为中心的上下产业链配合的很好，IDC提供电力、机柜和带宽服务，硬件厂商提供基础设施，我们提供云平台技术，上面的PaaS或者SaaS厂商提供相应服务，云生态和谐共存。

InfoQ：如果部署在客户那里的托管云平台系统需要升级，对客户的服务是透明的吗？

程辉：保证部署在客户数据中心的托管私有云无中断地平滑升级是我们的核心能力之一。面向大规模业务的互联网分布式IT基础架构一个最重要的特点是不允许中断。以微信为例，用户基数很大，几乎每分每秒都有人用，微信从上线到现在，几乎每天都有很多变更，但不能中断服务。云计算也是这个道理，客户把服务交给我来管理，我需要既保持稳定又要不断的改进、变更和升级。为了保障无中断升级，我们推出了很多举措，比如，我们在升级的时候，会给客户的业务做热迁移，保障业务连续性，用户几乎感觉不到服务中断。通过这些手段，每次OpenStack推出新版本时，我们都能及时跟进，现在我们公有云和所有的托管云客户都是运行在最新的OpenStack Juno版本上的，我们为客户提供托管的OpenStack有一年多了，都是从早期的G版本一路升级过来的。既然我们做托管云，也需要按照最严格的公有云标准来要求自己。

InfoQ：分享下你在开源方面的心得吧。

程辉：这需要从我在新浪工作时说起，当时我没有做开源，接手的任务是把公司的云平台尽快上线。我招了一批在校实习生，让他们两个月之内不参与任何公司的内部工作，只在社区中做，找bug，然后尝试修补。如果提交的补丁不规范，就会被社区退回来，有人曾经被打回20多次，通过这个过程，社区帮我很好的培养了这些人。在新人成熟之后，云平台只用了一个月时间就上线了。后来，我们被邀请去国外分享经验，我也有了创业的原始动力。后来就成立了UnitedStack，即使在资本很紧缺的情况下，我也会安排工程师全职在社区当中做。正因为如此，我们的系统稳定性才会很高。

另外，社区的架构设计和文档对我们很有借鉴意义。比如，某一个开源的账号体系，开始我们认为特别复杂，设计了几十个新的概念，不可思议。但是，后来我们在设计云

平台的账号系统时，才发现人家的设计是多么好。如果没有社区经验，是很难设计出来的。通过社区让我们知道了这些东西，让云服务产品更加有竞争力。

InfoQ：你认为 UnitedStack 的核心竞争力是什么？

程辉：刚才我已经说了一些。第一个是开源，目前在中国市场主流的云当中，我们算是唯一一个完全基于开源来构建的商业的生产的云，我们目前云系统采用的两大开源平台，OpenStack 和 Ceph，不仅开源平台为我们提供了源源不断的动力，我们还有一批非常懂开源的工程师，保证我们团队在开源业界的领先水平。第二个是互联网精神，既要变又要稳。公司核心团队基本上来自于互联网公司，因此我们有能力将互联网的基础设施和运维管理经验带到客户的数据中心。第三个优势，商业模式的创新，我们是国内第一家旗帜鲜明地提出托管云理念。如果对明年或者后年的云市场做一个预测的话，托管云会成为一个不可小觑的云计算细分市场。

InfoQ：你对目前云计算的发展现状有什么样的看法？

程辉：中国云计算市场现在还没有清晰的市场区分，总体发展还处于初创和混沌期。具体表现在，目前主流的云服务产商均采用的是自研的私有技术、私有 API，云平台之间没有统一的互通接口，缺少统一标准，无法通过标准参数来衡量一个云服务的优劣。

基础设施云计算技术，不论是 IaaS 还是 PaaS，大约未来 3~5 年左右时间会成为高度商品化的技术，商品化意味着花钱就可以买来，有市场有技术，而且市场和技术可以交易和转换，到那个时间，云计算市场竞争将从技术竞争真正转变为资源和服务的竞争。

比如，我们提出的托管云服务其实对应国外的是 Managed Private Cloud，这在国外

是一种主流的私有云交付方式，不论厂商、企业用户还是媒体都非常清楚。

InfoQ：云计算市场有哪些细分领域和玩家？他们分别有何特点？

程辉：我就按大家最常见的理解分为公有云和私有云两大体系。公有云市场按平台技术类型来看有两大类：

第一大类是基于自研的私有技术的公有云，比如阿里、腾讯等互联网巨头提供的云平台、外资的云（如AWS, Azure）、Ucloud，青云为代表的创业公司的云；

第二大类：基于开源技术构建的公有云：如京东云、金山云，UnitedStack、还有电信、联通等运营商的云平台，都是基于开源的OpenStack平台构建；

云计算和其他行业一样，顺应从闭源技术到开源技术的发展趋势，我们看到，2014年之后新成立的云平台，基本上都属于大二大类，基于开源构建。

云计算是可以OEM的，透露一下，到目前为止，国内已经有接近10家IDC、互联网公司公有云厂商的底层是Powered By UnitedStack的，即我们团队为其提供完整的公有云平台、技术还有运维服务，初步实现了IaaS云平台的商品化。

私有云有目前非常明显两大体系：

一类是商业VMware生态，目前私有云市场占有率非常高，尤其是在传统行业，但是目前大量只解决了虚拟化的问题，分布式存储、SDN网络等云计算核心技术还很难应用起来。

第二类还是OpenStack开源私有云生态，目前OpenStack开源私有云模式已经被广泛接受，在VMware最稳定的、市场占有率最高的金融和政企行业也可以看到越来越多的应用案例。UnitedStack的OpenStack私有云方案已经帮若干家金融和银行公司替换掉了VMware解决方案。

InfoQ：按照以前的IT规模，可能是市场成熟之后，有两三个比较大的卖家。你觉得云计算这个市场，会遇到这个问题吗？

程辉：不会例外，也会是这样的，大者恒大，因此，我们在未来两年必须变得强大起来，否则就会被淘汰出局。

InfoQ：UnitedStack在未来几年的路线图是什么？

程辉：技术路线上，我们会坚持开源，投入更多资源将开源项目产品化。在基础设施服务层面，高性能SDN网络和高性能统一存储将持续是我们的重点。SDN网络在开源界也是最近两三年才开始逐渐被关注和被应用起来，目前已经初步实现了SDN网络的构想，但其性能和稳定性还有进一步提升空间，在我们的计划中，未来1年，SDN网络的性能还有3到5倍的提升，并且会新增更多企业级安全特性，进一步满足严肃的企业级应用。

高性能统一存储的目标很简单，不仅要完美的替代传统的SAN企业级块存储，还能够为大数据、对象存储等业务提供底层支撑。性能优化方面，目前我们的分布式存储读写IO延迟已经突破了1毫秒，几乎接近分布式块存储的极限。在提供极高性能的同时，我们还在数据安全性方面下了很大努力。今年会继续在存储多样化上下努力，比如，刚刚上线的NAS存储服务和虚拟SAN功能，在行业内也是独一无二的。

基于扎实的基础设施架构，我们还将在PaaS层构建更多服务。

首先是容器技术的大规模商用。UnitedStack是国内第一家提供容器服务的云服务厂商，今年将在Docker存储和网络方面做一些功能优化，解决目前阻碍Docker容器服务商用的问题；

其次，将大数据与统一存储做整合，将OpenStack云平台和Hadoop大数据平台两大开源体系合二为一，真正实现我们内部早年提出的“一个底层，多个平台”的构想；

第三，将持续引入更多的开源的和商用的PaaS层服务，比如MySQL,Mongodb, Oracle数据库服务，Redis, Memcache等缓存服务，让开发和运维变得更容易。

受访者介绍

程辉，UnitedStack公司创始人兼CEO，曾任OpenStack基金会董事，是中国OpenStack领域最活跃的布道师和实践者。他致力于打造OpenStack全球生态圈中最成功的开放云平台：他创立UnitedStack是中国最专业的OpenStack开源云计算公司，拥有中国最强大的OpenStack开发团队，第一次真正将OpenStack开放平台的精髓注入了中国云计算领域。

文化篇

云适配陈本峰：HTML5 跨屏前端框架 Amaze UI 的开源之道

对陈本峰的采访，源于技术圈内的一个饭局，虽然大家对他的云适配创业经历很感兴趣，但是他却在自我介绍中反复提到了“开源”和“Amaze UI”，言谈举止中透露着对国内开源社区发展的关注和热情参与，特别是他领导的Amaze UI开源项目在正式上线2个月之后，在Github上就取得了1000多个star的关注度，这样的成绩在国内屈指可数。于是，便有了对陈本峰的采访和下面的内容。

Amaze UI 的前世今生

在中关村创业大街旁边的办公室中，陈本峰欣然接受了InfoQ中国总编辑崔康和高级编辑郭蕾的专访。陈本峰首先谈起了Amaze UI开源的背景，云适配是专注前端技术的公司，在把网站从电脑版转到手机版的时候，需要一个手机的前端框架。从创业初期开始积累，Amaze UI逐渐发展起来。

早期的时候，对这种移动端的Web界面框架比较少。国外的这块移动版做得相当的好，我觉得他们兼容性做得不错，但是速度非常慢，打开非常慢，所以无法忍受。于是我们当时决定自己做，所以就有了这套框架（Amaze UI），一直以来都是给云适配整套体系用的，后来我们觉得这套东西可以剥离出来，让别人也能用。所有人开发移动网站的时候，都应该需要这么一个框架，因为这样可以大幅度提高开发效率，不需要重新发明，这是一个基本的思想。

在开源之前，陈本峰的团队主要做了3件事情，这些看似微不足道的步骤为开源项目打下了良好的基础：

- 代码精进化
- 文档规范化
- 启动内测

国内开源项目尚处萌芽期

随着国内技术社区的发展，国产开源项目越来越多，但真正运营成功并取得广泛关注的例子并不多。在笔者抛出这个问题之后，陈本峰显然已经早有答案。他指出，从国外的开源经验来看，一个项目要想成功，必须有一个专职的研发团队来做。虽然我们谈开源，经常说靠社区的力量，但是最核心的推动力还需要是专职团队，并且这个专职团队是真的为社区服务的。

国内很多开源项目，大多是开发者自己的兴趣爱好，并不是公司层面来支持项目的，经过一段时间之后，开发者工作调动或者公司业务重心转移，就会导致项目夭折了。陈本峰分析过Github上的一些开源项目，刚启动的时候，更新频率很高，一个月有几十次，但是到后来，基本两年都没有更新一次了。这种状况无法给潜在的开发者信心。

对于开发者来说，评估开源项目可用性主要有两点，一是社区支持度，二是活跃度。这两项不达标，就没人敢用。陈本峰认为，如果开源是一个公司行为，而不是个人兴趣爱好，那么活跃度可以有保障。

如何成功做好开源项目

即使公司提供支持，开源项目就可以成功了吗？显然不是！陈本峰强调，开源要用一种服务社区的心态去做，而不只是服务自己公司内部。

虽然 Amaze UI 是云适配整个大产品体系的一部分，但是开源之后，Amaze UI 团队就只需要去考虑外面社区的开发者的需求，把我们自己的内部团队也当成跟外面社区等同的客户去对待，不因为是内部的团队就优先处理，而是看这个需求到底是大家普遍认为一个比较需要的高的需求，还是它就是我们的一个特殊行为。如果是我们的一个特殊行为，就应该慎重。让我们内部的团队基于开源基础上自己去改，改完以后由开源项目团队判断是接受还是拒绝掉，完全独立地由两套团队去运行，他们的目标也就非常明确。云适配跟 Amaze UI 开源的目标是各自明确的，互相独立的。我们在做的过程中，也发现外面社区有很多比较强烈的需求，比如我们在做 Amaze UI 里面发现最大的一个需求就是除了 Amaze UI 本身提供的一些功能，客户需要一些第三方的英文插件。其实这个需求本身在云适配这块是没那么强的，因为云适配这块主要还是针对移动端，而客户需要的英文插件是开发者在做一个横跨手机、平板、PC 三屏的网页时会需要的。并不是云适配的强烈需求，但是在我们的 2.0 阶段最重要的工作就是做这件事情，这就是个非常典型的例子，说明我们是优先考虑外面开发者的，然后才是考虑内部开发者，或者就等同对待。

这并不是纸上谈兵，Amaze UI 项目如今由两位全职的开发者在推动和维护，都是云适配的员工。

用产品的标准做开源，解决移动化难题

对于开源项目的定位问题，Amaze UI 虽然提供英文的相关介绍，但是它更是为国内开发者优化的，陈本峰分析了几个原因：

- 本土化的支持，Bootstrap 没有做专门针对中文的支持。字体是网页里面非常重要的一页，直接决定了网页展示出来的体验的好坏。Bootstrap 里面是没有定义中文字体的，这就会导致每个浏览器都会根据默认设置去选一个字体。比如说 IE 在

XP和Windows 8下字体就是不一样的，在苹果下面字体又不一样了，然后在各种手机浏览器上面字体都不一样。最后导致做出来的网页可能在各种浏览器、各种操作系统下面看起来效果都不一样，是完全不可控的。而且可能会导致排版格式变掉。Amaze UI里面很严格的定义了中文字体，做到在各种设备、操作系统、浏览器下看到的效果基本上是一样的，比如说中文字体，我们用的是雅黑。在Windows底下是雅黑，但是在苹果底下是没有雅黑这个字体的，那我们就用最接近的黑体去做。Bootstrap基本上是用13号字，我们是用14号字，字号的大小也会导致排版不一样。Bootstrap不太可能去加中文字体，因为如果一旦加中文字体，就还要加日文字体、韩文字体、法文字体，Bootstrap就会变得巨大无比了，这肯定也不是产品设计的初衷。还有对本土浏览器的支持，当时做Amaze UI的另外一个想法源于浏览器的兼容性，对于多数前端开发者来说，或者都是一个梦魇。可能开发一个网页用一个月的时间，但是做浏览器的兼容可能要花两个月的时间，甚至都做不完，面向的浏览器太多了。这些工作没有必要让开发者一遍又一遍的重复。所以当时我们想做一个开源产品，大家基于这个产品，把浏览器兼容性都调整一下，以后使用这个产品就行了，节省了大量的工作。国内浏览器和国外浏览器也是很不一样的，像360、搜狗等，而且国内有双核浏览器，这也是国外不存在的。针对这些中国特色，我们会做一些调整和一些特别的优化，这也是我们跟Bootstrap一个比较大的区别。

- 移动优先，Amaze UI一开始为移动端开发的，所以非常考虑在移动端的表现，要让整个代码体积尽量小。另外尽量的采用CSS3的动画，动画效果以前在PC端，都是用JS版做，一方面要下载JS代码。另外一方面是它对机器的要求比较高。因为JS需要大量的CPU运算。移动端的浏览器，相对来说都比较现代，都是支持HTML5的，使用CSS3动画就会节省代码，因为一行CSS3代码可以解决一个动画的问题，代码体积会小很多。第二，因为CSS动画是浏览器原生支持的，所以会有硬件加速。硬件的运算能力是要比用JS软件上的能力要强很多的，所以整个移

动端的体验会好。

• 组件化，Amaze UI非常强调组件这个概念，近两年有一个非常热的技术叫做Web component，它的意思是说，网页的每一个构件，都可以封装成一个组件，这种技术在后端已经非常成熟了。比如说Node.js里面的NPM，可以用来管理各种各样的包。Ruby的Gem，Python也有，但是前端是没有的。在前端大家的做法是非常原始的，比如要做轮播图，就是拷贝大段的Html5、CSS和JS，这个很大的问题就是，只要拷错一行，就不工作了。另外一个是更新的问题，来源的组件更新了，本地的代码也需要更新，可能一个修改就直接导致更新不了。所以前端整个开发的技术还是相对比较原始的状态的，所以Google在2012年的时候提出了Web component的概念，这个概念发展的非常快，W3C已经把它列入标准的开发范围了，现在已经在推进这个Web标准，我们设想未来的Web前端开发，应该是基于这种组件式的，所以我们也做了很多组件。Bootstrap并没有朝这个方向去走，它更多的是强调它这种框架的底层架构，而我们强调组件。而且这种组件是非常具有本土化特质的，比如说我们上面有百度地图的组件，国外用Google。我们的客服的组件，都是第三方部分提供商，或者一些视频播放的组件，视频播放组件可能会播放土豆优酷的视频，在国外会用Youtube。未来我们希望做成类似于Node.js的NPM的包管理的系统，程序开发者需要什么组件，一个命令行直接就下载下来了。

说到Amaze UI对待开源的认真态度，版本路线图的规划也是一个例子。陈本峰介绍说：

我们是比较严格的按照国外比较先进的开源项目的运营方式，比如说我们会找Bug，去分级，分成P0、P1、P2、P3、P4。P0是前面要解决的，P1会在下一个发布版本会解决，P2 我们会在下一个版本有时间的条件下去做，没有时间会往后推。P3 属于这种未来可以讨论头脑风暴的，用户提交上来一个Bug，一个issue，我们马上就会做一个级别的判定，这样子提这个Bug的开发者会知道，他的这个问题大概是会在什么时间

阶段会被解决，而不是说就是大家提上来了，我们把所有的 Bug 拢在一起，而是清清楚楚去把问题分类，确定会在哪个版本解决。版本规划也是，每一个版本的工作重点都分的很清楚，有很清晰的规划图，清晰的 Bug 管理系统，让开发者觉得这个项目比较靠谱，认真对待的每一个版本，很认真对待用户提出来的每一个问题的，而不是含含糊糊的，让用户根本没有期待。

开源亦能双赢

目前国内很多企业做开源都是抱着试试看的态度，那么开源仅仅是做活雷锋吗？陈本峰认为从商业角度，Amaze UI 和云适配也是受益的：

- 本身云适配业务是让用户把网站转到手机端，所以我们对兼容性、适配性是非常关注的。但是单凭一己之力，是没法做到兼容性非常完善的。我们开源出去，如果这个产品有别人用，那别人也来贡献代码，这样也能够反过来帮助云适配这个产品，能够做到更好的兼容性、适配性，对我们产品的提升是有直接帮助的，所以我们愿意去做这件事情，也值得投入做这件事情。
- 做招聘，Amaze UI 的开发者全部是前端的开发者。我们去招人的时候，就在 Amaze UI 上打广告，大家会觉得 Amaze UI 这家公司前端工程师，是一个不错的选择，就会愿意来，成本跟猎头费的成本也差不多。从这点来说，对我们就是招人肯定是有帮助的。目前 Amaze UI 在招募技术爱好者，也欢迎大家参与。
- 对于公司这个品牌来说，如果我们做了一个全中国最流行的一个前端框架，那么云适配以后做跟网站相关的一些业务，肯定会得到别人的认可，这也是一个品牌上的关注。所以我们会去做这件事情。对于另外一家公司，他可能就没有这么大动力了，比如说如果是一家做电商的公司，他可能不会有那么直接的帮助，那招人直接花猎头费了，他也不做网站相关的业务。可能跟云适配自己本身这个特定

的业务是非常相关的。

开源的未来出路

开源项目有哪些出路？陈本峰分析了国外的例子：

开源项目参与模式现在在国外是比较成熟的，基本上国外 2B（To Business，面向企业，以下简称 2B）产品的公司基本都是做开源的，我觉得他们的这个商业模式有几种，一个就是做收费技术支持，然后就是做培训，我们在做的过程中已经有这种参与，已经有人找过我们去给客户做收费培训。技术支持也可以做，这是比较容易看到的。还有就是去做一些系统集成的解决方案的，像 IBM，IBM 做了这种大量的开源，像 Java、Eclipse，基本上做这种解决方案，当然解决方案里面利用最高的还是它的硬件。当 IBM 的软件不是主要收入来源的时候，他就愿意去做开源的软件，加上自己的硬件卖出去。像 Google 做开源的目的是通过开源这种方式促进各种人去使用互联网，越来越多的人使用移动互联网，他的移动搜索就赚钱了，为什么 Google 会去做一个开源的浏览器 Chrome？Google 的商业模式在于流量变现，只要世界上有越来越多的人上网，就有越来越多的流量，那他就能变现，这是 Google 的一个商业逻辑，那这些都是跟他支撑业务是有关系的，如果纯支撑业务没关系，那就是培训，还有技术支持，Redhead 就是这种模式。

谈到对未来的这种开源发展趋势的看法，陈本峰对 2B 市场的开源报以比较大的期望：

我觉得开源在国内市场里比以前是要好很多了，市场繁荣多了，今天的 Github 上面有越来越多的中国开发者出现了，随着这个行业的发展，未来开源这个事情会在国内会越来越流行。当然我觉得可能主力应该还是那些大的互联网公司，因为国外主力也都是像 Google、Facebook 和雅虎这些公司。现在还处在一个萌芽期，哪一天它真的能爆发，

就是看这些大佬们在这块开始发力的时候。可能也是因为在中国做 2B 的大公司是非常少的，国外这种 2B 的大的上市公司是非常多的。在中国整个 2B 的企业还没有完全起来，这个也限制它开源时期的发展，为什么呢？因为首先这个企业有很多内部系统的集成的需求，如果不是开源，他没法知道这个产品是不是跟他内部的现有的这些产品能够很好的融合在一起。那你开源之后，他自己先拿过来研究一下，是不是结合的好，所以这个时候，把产品开源，其实是一种变相的推广手段。第二个考虑到一些安全性的
问题，开源之后客户也可以消除对安全性的隐忧，像我们云适配也是，它也是针对企业，是个 2B 的产品，我们基本上也还是按这条路来走的，所以我觉得国内开源能够飞速发展，就是有两个条件，一个就是 BAT 一些大公司开始介入、开始投入，第二个就是国内 2B 的公司开始参与进来。

在本文发布之际，Amaze UI 发布了 2.0 正式版，感兴趣的读者可以访问其官方网站了解详情。

受访者介绍

陈本峰，云适配创始人 CEO，Amaze UI 项目的领导者，国际互联网标准联盟 W3C 中国区 HTML5 布道官，专注互联网标准以及浏览器相关技术研究超过 10 年。曾任职于微软美国总部 IE 浏览器核心团队，师从浏览器行业泰斗 Christian，参与 IE 的 HTML5 引擎设计开发以及下一代互联网标准 HTML5 国际标准制定。

3月10日，豌豆荚的张铎成为中国第5位HBase Committer。现在越来越多的公司和开发者都开始关注开源，开源正在经历前所未有的繁荣时期，但这只是开始。从整体比例来看，国内参与开源项目的人并不多，很多人都不知道如何参与开源项目。在这个背景下，InfoQ采访了张铎，希望能深入挖掘HBase项目组织架构、运营、流程等方面的细节。

58

中国顶尖技术团队访谈录

第一次提交代码

作为一个历史悠久的开源项目，HBase非常复杂，据统计，HBase差不多有34万行代码，主要使用Java语言编写，部分模块可能会使用C语言。我在2008年底的时候就开始接触HBase，但是提交第一个Patch是在去年的9月份，当时在工作过程中发现，HBase有时候会丢失数据，确认自己的代码没问题后，我开始怀疑是HBase的bug，定位问题后就立即对该问题进行了修复。而小米的冯宏华已经是HBase的Committer，也正好是我的学长，在冯的鼓励下，我把自己发现的问题提交给了官方。

提交后不到10分钟的时间，就有一位Committer联系我让改代码的说明格式（有些地方不符合规范），简单修改后，这位committer立即@了负责这块代码的其他Committer。整个提交过程非常快，HBase方的反馈也很好，和我之前想的根本不一样。

如果要总结下第一次贡献代码的经验，我觉得应该胆子大点，不要害怕。不要怀疑自己的能力，发现问题就提交。不要担心自己的英文不好，只要发现问题就往上提，

这对自己的成长也有好处。很多国外的程序员写的代码很烂，但是他们却敢于提交。而提交后又有很多牛人来帮你修改，这相当于免费请大牛当私人教练。

如何成为HBase的Committer？

并不是第一次提交代码后就能成为Committer，只要提交过代码的都是Contributor的角色。Contributor要成为Committer，需要持续不断的贡献代码，并且开发过新的feature（猜测）。当代码贡献到一定程度时，项目管理者会主动与你沟通是否愿意成为Committer，并不需要自己申请。

成为Committer之前，我一共提交过30次左右的代码，从官方的邀请信中可以看到，PMC（项目管理者）比较看重我为HBase 1.1提交的几个大的feature，以及对HBase整个项目的单元测试流程的改进贡献。

目前HBase共有41个Committer，但真正活跃的不到一半，目前也没有相应的退出机制。Committer之间主要是通过邮件沟通，没有固定的在线沟通时间。

代码提交后的流程

Contributor不能直接push代码到仓库，他的代码需要经过Committer审核。如果是一个简单的改动，那一个Committer就可以直接做主。如果是一个比较大的改动，那就需要多个Committer一起讨论才能决定。如果是可能影响兼容性的改动，那需要与该版本的负责人讨论后才能确定。

HBase的代码并没有使用GitHub进行管理，GitHub上的代码只是一个镜像。Apache基金会中一些比较老的项目使用的都是官方自建的Git仓库，缺陷管理工具用的是JIRA，提交代码后，JIRA中相关issue的状态会变为Patch Available。同时，项目机器人会定时扫描是否有Patch Available状态的issue，如果有，它会下载相应的附件，并

通过脚本检查格式是否正确、单元测试能否通过，并把结果发回到 JIRA。当 Patch 要合并到某个版本之前，该版本的负责人会重点进行测试，包括功能和性能上的。

HBase 的项目架构

HBase 的项目直接负责人称为 VP，VP 全职负责这个项目。VP 下面是几个 PMC，每个版本的 release manager 一般是某个 PMC。PMC 下面就是 Committer 了。一般情况下，一个 Committer 会对 HBase 的固定负责某个版本的某几个块功能模块特别熟悉，所以当 Contributor 提交代码时，项目组是有审核该代码的第一负责人。HBase 项目组会优先让对该模块比较熟悉的 Committer 来审核，这个 Committer 的意见也是最重要的。

HBase 主要的代码贡献者都是 Cloudera 和 Hortonworks 的员工，他们都是全职负责 HBase，甚至 HBase 中写文档的人都是 Cloudera 的员工。Cloudera 和 Hortonworks 都是基于 HBase 的商业公司，他们同时维护开源的 HBase，但同时也都有自己的商业版本。

社区分歧

每个开源项目都会遇到技术方案分歧的情况，同样 HBase 也有。HBase 在这方面并没有好的解决方案，每次讨论这样的问题时都会分为两派，大家都在说自己的解决方案以及优势，但是永远也没有结论。目前也没有相关的投票机制，比如谁的得票多就听谁的，因为投票很容易导致产生更大的分歧。

其它开源项目也没有好的解决方案。如果社区比较融合，大家都抱着解决问题的心态来看待这样的分歧，那还好办。但如果大家都比较偏执，一派人坚持要这样做，另外一派人坚持要那样做，那这样下去稍有不慎社区就可能分裂，这已经有先例了。再或者就像 HBase 一样先挂起这问题，但也不是长久之计。其实分歧问题和社区文化直接挂钩。

开源谈

如果只靠个人无私奉献，开源项目很难发展起来，更不用说建立生态圈。如同公司一样，开源社区需要有人来推动、管理和运营，开源不仅仅是代码本身。比如前面提到的开源文化，这完全是需要有人来引导的。

豌豆荚一直比较崇尚互联网「开放」「平等」的价值观，也很支持重视开源技术。公司决策层领导层也非常重视员工在技术方面的长期积累，所以也允许我有一些比较自由的时间来研究 HBase，做一些贡献，也不要求这个研究马上得在多少天内就贡献多少代码或者做出多少新 feature。我觉得豌豆荚对于基础技术的长期积累还是很看重的，而且很有耐心。纵观一些好的公司，一定会多给员工一些属于自己的时间探索新的东西。如果每天都很忙，不停的在加班，那什么时候去进步了？员工的成长甚至跟不上公司的成长，长期来看反而会拖累公司。

【编者按】《开源启示录》是 InfoQ 推出的重点专栏，旨在通过新闻、文章、访谈、用户调查、迷你书等形式，报道国内外知名的开源软件以及开源的发展状态，并分析目前开源的现状，总结国内外企业以及个人在开源方面的成功经验以及失败教训。如果您对开源感兴趣，请关注《开源启示录》，也可以加入我们的 QQ 群（群号：319967710）参与讨论。

受访者介绍

张锋，豌豆荚基础技术负责人，目前主要关注存储相关的技术。2010 年研究生毕业，来豌豆荚之前一直在网易有道工作，从事的也是基础技术相关的工作。2014 年底带领团队开发了名为 Codis 的分布式存储解决方案，并于 2014 年 11 月 7 在 GitHub 上开源。

阿里巴巴研究员赵海平：从Facebook到阿里巴巴

赵海平，2007年加入只有不到50个软件工程师的Facebook，致力于软件性能和架构分析，在此期间创建了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。HipHop项目之后，致力于“用异步处理来优化分布式系统”的设计理念中，并为此做了多项分布式数据库的优化研究，在PHP语言中加入了yield和generator的新功能，来帮助日趋复杂的Facebook网页设计。

2015年3月，他回到中国，加入阿里巴巴技术保障部，任职研究员，将重点攻克阿里巴巴在软件性能以及Java使用过程中遇到的技术问题。

InfoQ：首先欢迎您回到中国。可以介绍一下您加入阿里巴巴的初衷吗，阿里巴巴最吸引您的地方在哪里？

赵海平：去年机缘巧合，我和阿里巴巴的同事有了交流的机会。当时我们聊了很多技术细节，发现阿里巴巴的规模非常之大，很多技术上的难题是美国公司都没有的。比如说双十一这个问题，没有哪家美国公司单天有这么大的交易量，这是很特殊的一个问题。这个技术问题对我特别有吸引力。

当规模大到一定程度，简单的问题也会变得复杂。有的时候软件就是这个样子，在一台或者几台机器上执行是一个情况，当机器多到一定程度时，对软件的要求就特别高了。在多台机器上，怎么才能保持很快的速度，并且节省机器，又不出问题，这是一个

很难的技术问题。

单天的资源需求是平时的好多倍，怎么计划机器，让峰值最高的那天不出现问题，平时又要做到很好的利用，这是很不容易的。我特别希望自己能够有这么一个经历，去阿里巴巴解决这个问题，这是在其他公司找不到的技术问题，而且跟我很对口，我一直在做的都是怎么提高大规模系统的性能、稳定性，所以这正是我的兴趣所在。

InfoQ：您在阿里巴巴的新角色就是解决这些基础设施的性能问题？

赵海平：基本上是几个方面，性能、稳定性、容量、架构，还有运维，恰恰就是我现在这个团队——技术保障部——的工作。性能提升上去，容量就增加了，随着我们监控系统的改进，系统的稳定性也会提高，运维也会更方便。如果发现架构上的问题，我们也会做些调整。

InfoQ：谈到性能问题，定位是很关键的一点。像这种规模的分布式系统，如何实现全系统的监控，准确定位问题就非常重要，您会在这方面发力吗？

赵海平：Profiling特别重要。如果能有一个特别强大的Profiling系统，就知道整个系统在哪个地方，哪台机器上，花了多少CPU、内存、磁盘IO或者网络带宽等资源，才能知道优化什么地方效益最大。

所以我的第一步工作就是帮助完善阿里巴巴的监控和Profiling系统，希望能够很清楚地把软件的整性能展现给大家，做到实时监控，同时让研发人员看到自己的代码在线上的运行情况，了解这些代码花掉了多少资源，这样有问题的话他们可以自己解决。

InfoQ：大家对您的最初印象多是来自HipHop for PHP这个项目。像

淘宝之前就从PHP切换到了Java，而Facebook选择了自己改进PHP。可以谈一下这个项目吗，当时的出发点是什么样的？

赵海平：HipHop也是一步步慢慢建立起来的。最初是我遇到了一个PHP的函数，在C++里也想用。当时想，重写一下就可以。不过那个PHP函数不断在变，我就想写一个简单的工具，把这个函数转换成C++，这样就可以跟上PHP代码的变化。那时正好机器开始吃紧，大家意识到PHP的速度问题，CPU消耗很大。大家就开始讨论如何提高PHP的性能。当时想法很多，有人想改变PHP本身，有人想干脆用Python或Java重写网站。

当时也重写过，有三四个人在做这件事情，但这些人改的速度远远赶不上另外二三十人写新PHP业务代码的速度。所以我们就想到写一个工具，来转换这些新写的代码，既不干扰既有的开发节奏，又能大幅优化性能，跟上变化。

当时我也读了下Zend Engine的代码，研究PHP为什么会慢。发现PHP速度之所以慢，是因为有很多的函数调用是动态的，而像C和C++里，很多函数是静态调用，不需要在执行的过程中去查询函数指针在什么地方，所以速度才快。

所以我们做了很大的调整，一定要改变这种方式，争取让所有的函数调用都能尽快实现，在编译的时候静态处理，执行的过程中就不需要再查询，指针已经在那儿了，这是最主要的加速思路。

那时候就萌芽了一个想法，如果能够把PHP直接转换成C++，也许这个性能问题就解决了。然后就花了很多时间去做原型。我们做了很多工作，把底层的PHP实现都改变了，有一个自己的Runtime Library，再就是一个PHP的扩展库，这个实际上是很大的一块代码。在这个上面，我们又写了一个把PHP转换成C++的一个编译器。先将PHP编译成C++，然后靠底下的这个库实现功能。这是最早期的工作。

不过这在当时只是一个副业，因为不知道这个东西到底有没有意义，是不是能提高性能。大概能拿出30%~40%的时间做这个。做完之后发现效果很好，就加入了其他同

事一起做。后来速度不断提高，第一年提高了2倍，第二年又提高了2倍，后来提高到5~6倍的样子。

现在回头看，如果当时雇很多人把网站改成Java的，也是可以做到的，但Facebook的发展可能要停半年到一年时间，甚至更久，就有可能对Facebook的发展带来不可预期的影响。这件事情主要还是业务推动的。

InfoQ：后来HipHop发展成HHVM，从原来的静态编译变成了动态的JIT机制，您也参与了这方面的工作吗？

赵海平：引入HipHop之后，我们也有自己本身的一些问题，比如产品环境和开发环境就是不一样的，这样多多少少会存在一些问题，也就容易出现bug。再就是Facebook的代码量非常庞大，编译时间非常长，另外生成的二进制文件也非常大（超过1G），发布也很困难。

这时就出现了HHVM。HHVM不再是把PHP转换成C++，而是采用了一种新方法，把PHP转换成一个中间码，这个中间码在执行过程中再转换成机器码，不过调用的还是我们原来为HipHop写的底层库，它取代的主要的是把PHP编译为C++的过程。

我并没有参与HHVM的编写，当时我已经离开这个小组了，另外一件事情吸引了我，这就是异步处理在分布式系统中的优化作用。

之所以离开这个小组，原因大概有几个方面：一个是，个人认为HHVM不再能把性能提高更多了，后来也确实如此，两三年之后HHVM出来，速度并没有更大的提高，最高只比原来静态编译机制快10%~15%，而且是因为静态编译这一块不再开发了。再就是新的课题特别有意思，具体我会在QCon北京上分享。

这就是为什么去GitHub上看，HHVM里面有我的代码，主要是底层的代码还是基于HipHop的。HHVM的头两个字母就是来自HipHop，引擎还是原来的，不过上面做了很多工作，把PHP转换成中间代码，这个有点像Java的JVM。这样的好处就是研

发过程和产品过程其实是一样的，而且不会有原来的那种超大二进制文件的问题。中间代码很小，PHP可以直接发布到线上系统上。

InfoQ：国外一些著名的互联网公司，在性能调整和优化的过程中，慢慢都发展出了自己的编程语言，像Facebook设计了Hack语言，Google有Go和Dart等语言，Apple有Swift等。这方面您有什么感想吗，Facebook的Hack语言您是不是参与设计了？

赵海平：Google的Go语言挺有意思的，写得非常好。Hack语言我没有太多的参与。

PHP是弱类型的，这是性能提高的一个瓶颈，而强类型的话就可以做很多优化。最初我们是想增强PHP的类型系统。强化数据类型，这是引入Hack的一个主要因素。

InfoQ：PHP7最近也有很多改变，性能提高也比较大。

赵海平：这也是我临来之前刚刚听到的。PHP核心开发组的力量是很强的。我也跟他们的人员交流过，他们对整个PHP的优化有自己的思路和想法，也做了很多工作。这是件好事。其实重要的并不是说哪个团队或小组把PHP优化到什么样的地步，有几个小组都在做这个事情，彼此竞争，会促进整个PHP生态系统的发展。这种竞争也恰恰说明了PHP的重要，所以会有很多人关注它的性能优化。

InfoQ：您对公司发明创造自己的编程语言怎么看？

赵海平：编程语言这个问题，我说两点。第一，不能把语言当成一个特别神圣、至高无上的东西。语言也是整个软件系统的一部分，只是它分割的很好，独立出来了，可以执行更多的功能，我们可以用它实现我们的功能，可是在整体架构上看，语言也只不

过是软件系统的一部分，而这一部分我们完全可以做一些调整，使其更适合我们的系统。而设计语言时一般考虑的是比较通用的目标，我们拿来用，只是因为它的绝大部分特性都是我们想用的。比如我们用C和C++写程序的时候，每次都要思考内存的模式是什么，是不是用share pointer，是不是用自己写的Object，内存的allocation/deallocation怎么做，一个语言帮我们把这些事情都做好了，这就是它的好处。它提供了一个很大的库，提供了一个软件执行的环境和范围，而这正是我们选择语言的初衷。

第二，作为一个公司来讲，不能说为了研发一个语言而去研发一个语言。这是没有意义的。一定要根据自己想要做的事情，在现有的软件架构当中，我们发现当前所用的语言提供的环境和范围不太适合，或者说这个语言的很多假设和假想，和我们所期待的东西并不一样，只有在这个时候，我们怎么也找不到一个合适语言的时候，我们才会创造一个新的语言，让这个语言更适合公司的事情。如果可以通用化，提炼出来，那就是语言，否则设计成软件库就可以了。

这是水到渠成的，不要强求。像Google开发Go语言，我认为它有自己的考量，可能在开发很多分布式系统的时候，现在的语言写法上不太直观，或者速度不够快，所以才创造了这么一个东西。Go应该能解决公司内部的很多问题，否则是很难存活的。

InfoQ：您可以结合自己这些年的经验，给中国开发者的成长一些建议吗？

赵海平：在美国的时候，我跟Facebook的中国员工聊的很多。我给他们最多的建议就几条。

第一，一定要提高交流能力。咱们中国人，尤其是中国搞计算机的人，很多人有个不该有的特点，就是喜欢把自己锁在黑屋子里埋头干活，跟机器交流特别擅长，跟人的交流一窍不通。这样不行，我相信在中国也是这样的，你不但要把自己的工作，技术活要做得特别好，而且要擅长表达自己的想法，擅长在工作当中讲述做的是什么，怎么样

能够说服别人，怎么样能够跟别人在不伤和气的情况下，把问题解决好，这是很强的一个能力，而这个能力不是在学校里能学会的，是我们走向社会之后慢慢学到的东西，这个我恰恰认为中国的员工，尤其是在美国那样的环境，因为不是母语，可能处理得就不是特别好，有时说出来的话比较生硬，给对方的感觉不是特别好。

第二，中国人比较谦虚、内敛，讲究内涵，自己心里有的东西不表达出来。但是在工作中，可以适度强势一点，勇敢表达自己的想法。当然这个建议是基于美国多元文化的背景，在国内大家的文化背景一致，也许可以探讨最合适的沟通方式。

第三，掌握好英语，开拓眼界。我觉得在计算机这个技术里边能够非常了解英语，把英语的这个隔阂给去掉还是非常重要的。

我回来的时间还不长，等和大家接触多了，可能会有新的想法，目前就这几点吧。

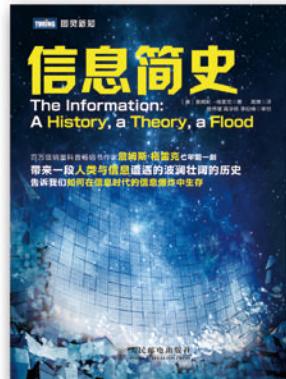
InfoQ：好，感谢您接受我们的采访。期待您在QCon上的分享。

受访者介绍

赵海平，资深技术专家。为Facebook第一个中国工程师，HipHop项目作者，并参与了HHVM项目，对编程语言以及分布式系统性能优化有着丰富经验。

图灵公司 始终以策划出版高质量的科技书籍为核心任务，推出了一系列高质量的畅销科技图书，是国内计算机图书最有影响的出版单位之一。

图灵社区 以为读者提供一流的内容为己任，拥有众多资深技术爱好者用户，是国内最专业的IT技术“交流+阅读”社区之一。图灵社区经过不断升级改造和尝试实践，已迅速成为国内知名科技电子书销售平台。



访问图灵社区 → iTuring.cn



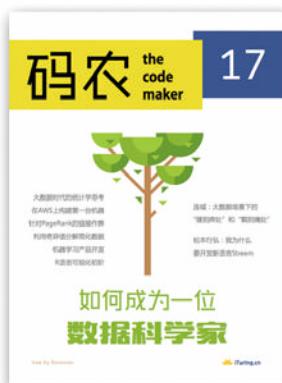
图灵访谈微信



图灵教育微信



图灵教育微博



码农的草帽底下，是一颗充满创造力的自由不羁的头脑。

码农们是勤奋的，加班加点的工作是常有的事情，城市夜间的灯火，有多少是在码农们的办公室和居所点燃？周末四处举办的技术交流和讲座，又活跃着多少码农的身影？线下读书，线上讨论，冥思苦想，动手实践，新技术驱动着码农们的脚步，码农们在改变着我们的生活。

读者俱乐部：218139230 (QQ群)

微博：@图灵教育 @图灵新知 @图灵社区

微信：图灵访谈：ituring_interview 图灵教育：turingbooks