

T O P T E C H T E A M

# 中国顶尖 CHINA

2016年 / 第七季

技术团队

访谈录



扫一扫，了解更多

Geekbang  
极客邦科技

InfoQ

Broadview®  
www.broadview.com.cn

## 专访吕毅： 链家网技术架构的演进之路

InfoQ就技术架构变迁对链家网平台服务架构师吕毅进行了采访。

## 专访蘑菇街七公：25倍增长 远非极限，优化需要偏执狂

InfoQ采访七公老师，看看蘑菇街是如何快速走出一条高效、实用的服务化发展路径的。



## 专访丁奇： 阿里云即将开源AliSQL， 超大并发、针对秒杀优化

即将开源的AliSQL又将为开发者带来哪些功能？针对于不同行业，AliSQL做了哪些工作呢？

## 华为开源数据格式CarbonData项目， 实现大数据即席查询秒级响应

华为宣布开源了CarbonData项目，InfoQ采访了项目技术负责人为大家解惑。

## 百度贴吧王伟冰：跳出历史技术栈， 谨慎寻找最佳优化思路

InfoQ采访了王伟冰老师，聊聊每一次架构变动背后的细节与故事。

## 中国顶尖技术团队访谈录 第七季

本期主编 韩 婷

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](mailto:feedback@cn.infoq.com)

商务合作 [sales@cn.infoq.com](mailto:sales@cn.infoq.com)

内容合作 [editors@cn.infoq.com](mailto:editors@cn.infoq.com)



## Mission

### 使命

整合全球优质学习资源  
帮助技术人 and 企业成长

## Vision

### 愿景

全球领先的技术人  
学习和交流平台



极客邦科技是一家 IT 技术学习服务综合提供商，旗下运营 InfoQ 技术媒体、EGO 社交网络、StuQ 职业教育三大业务品牌，致力于通过整合全球优质学习资源，帮助技术人 and 企业成长。

了解更多，请访问官网：<http://www.geekbang.org>



在微信关注 InfoQ



在新浪微博关注 InfoQ

## 关于 InfoQ

InfoQ 是一家全球性在线新闻 / 社区网站，创立于 2006 年，基于实践者驱动的模式建立，目前在全球拥有英、法、中、葡、日五种语言的站点。

软件正在改变世界。促进软件开发领域知识与创新的传播是我们的使命。每月有超过 200 万的技术人员访问 InfoQ 中文站。我们面向 5-8 年工作经验的研发团队领导者、CTO、架构师、项目经理、工程总监和高级软件开发人员等中高端技术人群，提供中立的、由技术实践者主导的技术资讯及技术会议，搭建连接中国技术高端社区与国际主流技术社区的桥梁。

## 2016 年重点关注技术领域

架构	容器	运维	云计算	移动	前端
大数据	研发	VR	IOT	人工智能	

## 产品

### 线上品牌专栏

架构、移动、开发语言、前端、容器、大数据、NLP、运维、云计算等。

### 迷你书

《架构师》、《云生态专刊》、《开源启示录》、《顶尖技术团队访谈录》、以及其他特刊及品牌迷你书。

### 新媒体

InfoQ 微信公众号、官方微博、垂直技术社群。

### 直播



### 垂直技术公众号和社群

聊聊架构、移动开发前线、大数据杂谈、细说云计算、前端之巅、高效开发运维。

### 技术大会

**QCon** 北京站 | 上海站  
全球软件开发大会

**ArchSummit** 北京站 | 深圳站  
全球架构师峰会

**GMTC**  
全球移动技术大会

**CNUT**  
全球容器技术大会

**ChinaTech Day**  
中国技术开放日

国内版：北京、上海、苏州、武汉、台北等 海外版：日本、美国、欧洲

## 卷首语 | QCon 大会主编 臧秀涛 Tom



金秋十月，在 QCon 上海 2016 的会场，新一期的《中国顶尖技术团队访谈录》又与您相遇了。

首先感谢您来到 QCon 现场。借此机会，先聊聊 QCon。

秋天是收获的季节，很多新兴的技术，最新积累的技术实践，正好可以拿出来做个分享。

InfoQ 一直致力于“促进软件开发领域知识与创新的传播”，而 QCon 就是我们的举措之一。每次 QCon，我

们都要挖空心思，邀请业界技术背景深厚的专家担任联席主席、出品人，一同来挖掘、筛选最有价值的话题，邀请最有实践经验的讲师。

每次 QCon 都会有很多不一样的地方。

比如，我们策划了很多新的专题，以前大受欢迎的架构、大数据之类专题策划思路也根据新的形势作了调整。

在线广告是很多互联网公司的主要盈利模式，广告系统的架构、策略、

优化都至关重要。所以这次我们专门策划了“互联网广告系统实战”这个专题，讲师阵容也非常豪华，专家分别来自 Google、腾讯、百度、小米、Spotify 等国内外最具代表性的互联网公司。希望参会者听完之后，能对互联网广告有整体且深入的理解。

像最近几年一直比较热门的大数据专题，我们就结合最新的趋势和热点，邀请到 Netflix 的架构师 Steven Wu（吴震），分享他们打造万亿级别数据流水线的经验。Apache Kafka 是 LinkedIn 贡献出来的开源项目，这次我们也邀请了 LinkedIn 数据基础架构部门 Kafka 组的架构师秦江杰，分享使用 Apache Kafka 进行关键业务消息传输的经验。

架构方面，我们也新增了微服务架构、业务系统架构等专题，不谈概念，只谈落地的思考与经验。

运维方面，这次将监控作为重点，Twitter 架构师黄浩将分享 Twitter 监控栈架构实践，届时可以看看他们的监控系统是如何处理十亿量级指标的。

另外，前端、移动、容器、Java、工程团队管理、技术创业和用户体验设计等多样化的专题，也能满足不同背景参会者的需求。

从前面介绍的演讲，读者可能已经注意到了，有很多来自国外的技术专家。没错，QCon 一直希望将国外的一些先进经验吸引过来，给国内的技术人员一些启发，进而促进国内外的技术交流。这次 QCon，来自国外，或者有国外工作背景的讲师有 20 多人。希望能给参会者带来不一样的体验。

参加技术大会，不单是听，还要多提问，找讲师交流。演讲是单方面的灌输，如果想将参会收益最大化，还是需要参会者主动去交流。

希望您不虚此行。

本期《中国顶尖技术团队访谈录》，集结了 InfoQ 对链家网、蘑菇街、华为、阿里云、百度等公司专家的深度访谈。

鉴前世之兴衰，考当今之得失。不管演讲还是访谈，希望这些内容都能对您有所裨益。

## 专访吕毅：链家网技术架构的演进之路



作者 木环

**吕毅**，链家网平台服务架构师，负责链家网多个公共服务。云计算浪尖时参与建设国内第一家 PaaS 平台 SAE，移动浪尖时作为初期成员与团队打造了国内 Top3 的 SuperAPP 手机百度，重度垂直时代加入链家网负责平台化服务建设。

链家网虽然成立于 2010 年，但是其技术团队却于 2014 年正式创立。此前技术开发采用的是传统模式，每个业务都会单独地重新开发，不仅造成各个模块孤立，并且开发人力投入成本巨大。鉴于互联网时代企业业务发展迅速，原

有的传统化方式已经不适用，链家网正式建立技术团队，在原有的传统架构基础上开始了优化工作。

团队对已有的业务进行抽象，将各个业务模块中的公共部分综合出来，据此添加了一层公共的服务层，实现了平



台服务化，扩展技术基础能力。此外，链家还重新搭建系统监控并完成日志监控，双级监控完善技术运营能力。目前的技术架构充分满足业务发展情况。InfoQ 就技术架构变迁对链家网平台服务架构师吕毅进行了采访。

### **InfoQ：谈谈您为什么会选择加入链家？**

**吕毅：**关于自我成长，相对于前两家我任职过的成熟互联网公司，在我入职时，链家网相对初期。这就预示着链家网这里会有很多挑战，我一直清晰认识到自己是有惰性的，所以我希望找到一家技术挑战多的公司，这样这些挑战会刺激我兴奋、鞭策我成长。在这里的一年，的确验证的当初我的这个想法，刚入职是我与两位伙伴做话务平台项目，不满一年，我们平台架构组已是 8 人的团队，负责着公司 6 项平台化服务，挑战与成长并行，大家都感受到架构设计、技术选型与实现、上下游沟通上突飞猛进的成长。

关于职业选择，我是技术角色，顺应之前的职业发展，我在抉择机会时候还是会选择互联网公司。然而，我将互联网公司分为两类，纯互联网公司与传统企业 + 互联网公司。在纯互联网公司这座大山里，我作为技术角色站在过最

好的山头，虽未登顶但也看清了这座山头的风景，转头看，最近些年才被关注的传统企业 + 互联网公司的这片山林，一片荆棘，而我具备的能力可以开垦这片大山，为了追求开垦的乐趣从而来了这里。至于为什么是链家，当时做选择时链家人为我描绘的链家蓝图吸引了我，当时了解到产品整体负责人闫觅之前是百度的高工，这让我也对链家好感增加不少，我想技术出身的 PM 在产品规划上应该会更理性些。也恰巧那会做选择时想咨询下前辈鸟哥意见，鸟哥说他下个月也去链家，这让链家的 Offer 加分不少。

入职前的种种对传统行业的顾虑，在入职后逐渐的全部解除了。这里提供的空间与自由度，管理的扁平程度，都是让我在入职前难以想象的。在链家网，时常在露台碰见闫觅、鸟哥，吐槽吐槽对产品的不快，对公司的发展的不理解，他们都会耐心的解释为何如此，从什么角度考虑做的决策，每次聊完后我都能自我纠正了想法，同时我也乐意将这些解答传递给我们的组员。链家网给予员工充分的空间，在保证自身工作的基础上，我可以与 HRBP 组织录制平台化服务系列课程、与员工发展部规划初级工程师培训、与外部技术社团联络到链



家来组织技术活动、与一些公司技术团队做技术交流都得到了大家的肯定与支持。这里有足够的空间，只怕你的想象力不够。

### **InfoQ：谈谈目前链家网的主要业务及团队规模？**

**吕毅：**链家网主要业务在很多公开资料中都有谈到，二手房方向是链家网目前的重心，围绕二手房开展的面向业主与客户的链家网、链家 APP，面向链家十万多经纪人的 Link 项目，都是链家网目前的重点业务。链家网目前员工 1000 人，产品技术角色占比达一半。

### **InfoQ：您提到链家网这两年在架构上做了很多的调整，可以谈谈链家网的架构演进历程吗？**

**吕毅：**链家网的架构演化的确从未停歇，从技术团队建立至今的两年时间，架构上大致分为两个阶段，如图 1 所示。

2014-2015 年技术上关注两件事，一块是将链家集团曾交付外包实现的面向经纪人业务改造为自主研发，另一块是从零打造面向用户的业务链家网与掌上链家 APP（目前已改名为链家 APP），即第一年的技术架构重在业务的建设。

2015 年至今，业务的逐渐成熟，引入了新的挑战。业务方向内的子业务细化伴随着链家新业务的开展，此时迫切

希望抽离公共技术部分，避免重复造车的同时也希望由公共服务来支撑好业务线发展，让业务线更好的满足产品迭代。从 2015 年开始，一系列的公共服务的建设，构建成了整体技术架构上的服务层，同时也扩展了公司基础技术能力，并推动着建设起了系统、日志级的监控。

### **InfoQ：您对目前的架构满意吗？下一步优化的方向是什么？**

**吕毅：**从技术架构的演化可见链家网成立的两年间，技术架构从无到有、从有到完善，一切都在快速的发展着。我们目前的技术架构是充分满足当前的业务发展情况的。要说对架构是否有不满，的确是有的。如今链家网业务各个方向发展迅猛，技术架构上的压力不小，理想状态是希望技术架构在支撑业务时，时刻游刃有余，但这会是条任重道远的路。

下一步的调整与优化方向，还是配合业务上一起做好服务化。业务层面希望将目前打包在一起的功能逐渐服务化，而技术架构上需要提前调整与优化，提供服务维护、服务治理、服务监控、服务通信等一系列围绕产品服务、技术服务的周边技术支持，这块是明确的方向。然而，一些未知的方向，还得与业务线、管理者们常沟通，不断摸索，按

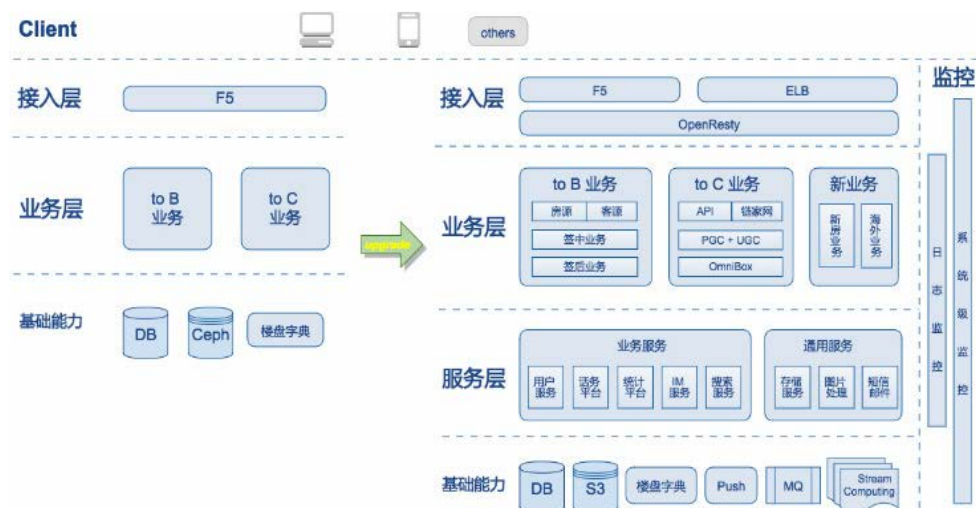


图 1

需计划着开展。

**InfoQ：服务层的优化已经进行一年多，这过程中有什么经验可以和大家分享的么？**

**吕毅：**在来链家网前到现在，算起来的确有一年多在做平台化、公共服务这些事情，伴随着自我成长，感慨的确很多，列三点深刻感受的在这里与大家共勉。

第一，服务源自需求。只有业务需要的才是值得做的，只有多个业务线都需要的，才是值得拿出来做成平台服务的。多去关注业务，寻找业务线研发团队共性部分，才能发现需求，通过与业务线的沟通才能发现做平台服务的价值，所以做平台服务的技术角色切忌闭

门造车，只有把自己当做半个业务线的RD才能感受到业务线的痛点。

第二，“第三选择”。常见行业内做公共服务的同学与业务线同学有些碰撞，双方各持观点，对一些边界问题拿捏不清。此时大家秉持第三选择，一起寻找更好的解决办法，目标达成一致，问题便可化解。当然第三选择的解决办法用在哪里都是合适的。

第三，服务的解耦。这是平台化服务的基本原则，服务间的解耦、服务内部的功能解耦，都是在日常设计、开发中需要注意的，只有将一块事情拆成多个点，才好做点与点之间的联系，以有限的功能点构建出无限的能力。这是老生常谈的点了，但依然值得再提一遍。

**InfoQ：请讲讲你们的日志分析平台架构吧？目前的日志处理流程是怎样的？处理的日志大概是怎样的数量级？**

**吕毅：**日志分析平台在前面技术架构图中，属于纵向的监控部分中的“日志监控”环节，主要解决业务模块、服务模块日志字段的数据收集、展示、监控。架构设计引用公开资料中的截图来说明（见图2）。

日志通过Kafka收集，根据日志所属RD配置的统计、监控规则通过Apache Storm实时分析日志，并将结果集数据存储数据库，实时分析期间若触发了监控规则阈值，则触发报警。数据

库中的数据可以做实时的数据展示，整套方案可以让研发、测试角色实时查看日志情况，避免了大家日常合并日志文件再做shell统计的问题，并提供平台可以持续使用。

目前日志平台每秒处理的30万行日志，处理结果的展示与报警延迟在2s以内，并且这套解决方案有计划在后续开源，让业内同学低成本掌握并构建到生产环境中。

**InfoQ：要设计一个高可用、高吞吐的日志平台，您认为需要重点考虑和解决哪几个方面的问题？**

**吕毅：**设计这样一套日志方案，有以下几点需要关注：数据的收集、数据

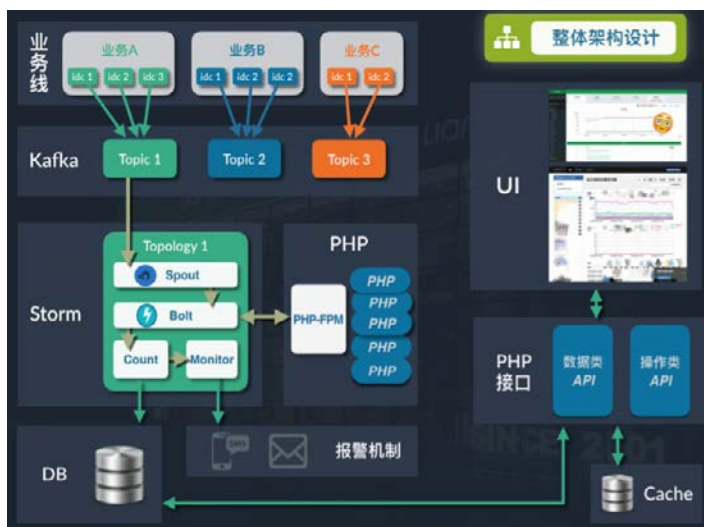


图2

的处理、数据的存储、数据的展示。

其中数据的收集部分行业内部不少用 flume 采集日志，我们最早期的 beta 版本也是如此，但在我们方案中缓存日志数据的组建 Kafka 在年初版本中发布了 Kafka Connector 功能，实现了类似 flume 能力，故我们上线前就改用 Kafka 全套解决日志收集问题。

数据的处理，因为要求实时性，行业内也有两套方案，Spark Streaming 与 Apache Storm。两者共性很多，但选择 Storm 的原因主要是 Storm 的设计与 Feature 更专注实时运算，而 Spark 做离线的大并发流式处理是不错的，例如流式批运算、图片处理，等等。

数据的存储，选型比较多，大家做 DB 选型时需关注企业级日志系统需要有大批量数据写入，特别是业务高峰时期，那么选择一个良好支持高并发写入特性的 DB 即可，我们使用的 HBase。

数据的展示，这块就比较灵活，根据自己需求，从 DB 中选择数据通过组织拼装成格式化数据，配合上前端特效展示即可。

**InfoQ：在你们的平台中，日志数据会采集哪些信息？你们是如何统一其他开发人员的日志格式和信息的？**

**吕毅：**如日志平台的架构图所示，

数据的采集基本是全量的，日志文件大部分是研发关注的信息，这些被关注的日志文件都会被收集，用作实时分析计数。

在日志格式这块，我们在接入层做了统一日志格式，故这份日志将会是所有业务的请求日志全集。对于业务模块、服务模块自己的日志，我们会给予建议，但没有强制规定日志格式，这部分有差异的日志格式，会在日志平台中研发角色配置统计规则时，通过正则匹配自助扣取想要的日志字段用作统计、监控。

**InfoQ：与日志监控相比，系统级监控对业务层、服务层的监控指标及目的是怎样的？**

**吕毅：**系统监控和日志监控比较，有这么几点差异。

**面向人群：**

系统监控，主要面向运维角色，RD 角色关注较少，RD 对于系统监控，更关注报警；日志监控，主要面向 RD、QA 角色，他们关注业务日志某些字段出现的频次、某些值的最大最小值等，同时也可以设置基于日志的业务报警。

**数据源：**

系统监控，更多收集 syslog、机器设备数据；日志监控，主要手机业务线自己打的日志内容。

两者互补，在需要了解业务所属的服务器信息时候，在系统监控上查看；想了解业务数据情况时，在这套日志监控平台上看。

**InfoQ：链家也是一家从传统公司转型为互联网公司的代表，你认为这中间，最大的挑战是什么？你有什么经验分享？**

**吕毅：**传统公司转型互联网，我认同链家集团老总发表的观点，这样的转型并不是将原有业务打死，全部线上化，而是传统公司加入互联网属性，是传统业务与互联网业务相结合的过程。链家网闫觅也曾说过“这是一次人文革命”，互联网更多的是一个工具，一个新型的商机渠道，一个提升效率的系统方案。

那么链家在做线上线下结合的过程中，最大的挑战应该有两点：传统业务的梳理与互联网化改造，线上业务与传统业务的融合。

传统业务的梳理与互联网化改造，链家网从创立之初就在践行，至今仍是重点工作之一，可见难度并非一般。在几次与经纪人业务侧的 PM 交流后，深感房产交易的复杂性，就北京而言交易中的一个环节可能会有 30 多种角色参

与，全国各个城市政策不一，更是难上加难，而链家网在努力优化流程，给业主、用户提供清晰明了的房产业务体验。这方面的挑战折射在技术上，就是复杂的业务流程控制、风控体系、数据安全等技术难点。

线上业务与传统业务的融合，存在于在 To C 业务与 To B 业务存在交集的部分。To C 业务是我们所长，但如何让还未进入房产交易流程内的普通用户，快速了解链家房产信息、交易流程，让普通用户在有需求时第一时间通过链家网、掌上链家 APP 发起沟通、交易，这都是挑战。折射在技术上就是如何保证好线上用户如何快速、满足需求的查找到信息、如何与经纪人快速建立有效沟通、如何沉淀意向客户并建立起关系等等的技术问题。

作为链家网平台架构组负责人，与组内 7 位伙伴一起，对技术追求极致的同时，我们也时刻关注着业务线的难点、痛点，希望通过技术手段提供支持与帮助，我想所有链家网角色的信念是一致的：让房产交易不再难，提供诚信便捷的服务体验！



## 专访蘑菇街七公： 25倍增长远非极限，优化需要偏执狂



作者 李东辉

**七公**，原名白辉，2014 年以前在阿里巴巴 B2B 主要负责 Aliexpress 资金中心、评价、任务中心等系统。14 年 8 月离开阿里出国游历，15 年回国后加入蘑菇街，目前在蘑菇街、美丽说、淘世界大集团共享的电商基础平台负责购物车下单小组及交易平台架构工作。

每秒能支撑的峰值订单数是衡量电商系统高并发可扩展能力的重要体现，在电商内提升每秒支撑订单数存在无数的方法，每一个方法都存在各自的优化角度和对应的障碍挑战，如何在一开始根据自身企业的特点选择最合适的优化路径，并其后在这路径上高效贯彻和执

行，对于团队尤其是初创团队而言都不是简单的事情。

蘑菇街技术专家七公，在 ArchSummit 全球架构师峰会分享了《跨越篱笆——蘑菇街每秒最大订单数 25 倍提升历程》，讲述的是蘑菇街从一开始只能支撑每秒 400 单的交易创建一直

到每秒 1 万单的优化实践内容，以及他们在这个过程中翻越的无数障碍和对应的解决之道。

现在我们就来采访七公老师，看看蘑菇街是如何快速走出一条高效、实用的服务化发展路径的。

**InfoQ：在 2014 年以前您在阿里巴巴 B2B 负责 Aliexpress 资金中心等项目，能否谈谈在阿里的这段经历给您之后出国游历、回国加入蘑菇街带来了哪些动因和影响？**

**七公：**我是 2011 年 3 月加入阿里巴巴的，彼时阿里整个的业务、技术体系都处在一个飞速发展的时期，我个人也获得了飞快的成长。受去看世界的梦想推动，2014 年 8 月我和女朋友一起从阿里辞职，出国游历了半年。

回来之后，阿里当时内部各方面整体已经趋向稳定，我希望找到另一家飞速发展的公司，北京上海找了一圈不满意，最终还是回到了杭州，加入了蘑菇街。在蘑菇街一年的时间里，我们业务和技术都获得了火箭般的飞跃式前进，我一年里的进步和收获比在其他公司呆两年得到的还要多，实现了和公司的共进步、同成长。

**InfoQ：您加入蘑菇街电商团队后带领团队同学仅用一年便完成服务架构**

**的各阶段升级，能否谈谈你们是如何规划优化路径并高效实现的？在高效率工作上你们有什么经验或技巧可以分享？**

**七公：**第一个阶段【蘑菇街系统拆分 & 服务化 1.0 体系构建】，是我们做 PHP 全面转型 Java 体系、初步建成电商基础服务中心的战略规划，在面临不停歇的业务需求和巨大的系统改造压力下，我们采用瀑布流工程方法，通过规划、分析、设计实现、测试、服务产品 & 文档交付的过程，高质量地把第一阶段的服务化建设根基像打桩一样打牢，然后通过进一步的迭代开发不断完善。

第二阶段【购买链路核心服务的性能提升 & 服务架构 1.5】和第三阶段【服务 SLA 保障推动稳定性提升 & 服务架构 2.0】，实际上是业务迅猛发展、流量不断上涨、日常和大促稳定性保障的强烈需求推动我们自身服务架构的升级。我们通过引入 Scrum 的敏捷开发模式，项目中的每个人都是猪（敏捷开发中，猪为项目负责人，鸡只是普通参与者，寓意来自猪要牺牲才能提供食物而鸡只要下个蛋就行了），每个人都要为服务框架升级和项目进展负责。

我们先后有十人以上共同推进了服务框架负载均衡、降级限流、连接切换优化等基础框架演进，以及监控、服务



端超时控制、多版本、分组隔离、动态路由等服务治理体系完善。

总结一下：高效推进上，我们的经验是首先要采用项目的方式进行管理，再一个是时机不同、阶段不同，选择的项目工程方法也不一样：

- 新架构探索：建议采用传统的瀑布式和迭代式开发；
- 架构的持续迭代：可以尝试一些新的开发方法。

**InfoQ：能否介绍蘑菇街系统架构的中间件系统？蘑菇街的一系列中间件是如何实现 Web 应用层和基础设施层对接的，各中间件如何确保在全站稳定落地？**

**七公：**蘑菇街购买链路服务架构示意图 1。

从蘑菇街购买链路服务架构 1.0 到 1.5，我们完成了服务化框架 Tesla、分库分表 TSharding、分布式缓存 MoguCache、Corgi MQ、配置中心 Metabase、调用链跟踪 Lurker 等一系列重要中间件的自研、完善和在全站落地。这一系列中间件的研发，是随着业务系统和服务框架的不断发展（最早我们只有 Tesla 一个中间件）而逐步衍生出来的。

2015 年初，蘑菇街全网还在 PHP 体系，Tesla 框架诞生后，我们先构建了用户、商品、交易、促销等一系列的电商基础服务中心：

- Tesla 框架支撑了 PHP Web 应用到服务中心的 PRC 调用场景；
- 服务中心内部有数据容量提升需

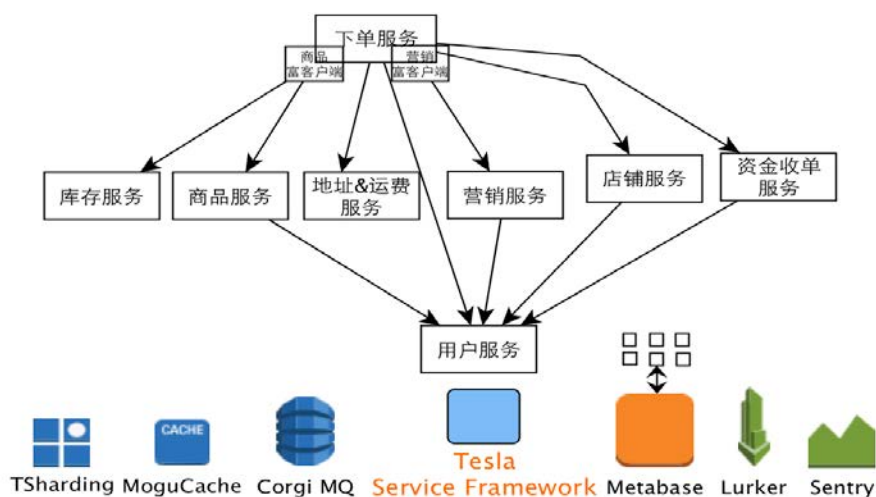


图 1

求，所以先后有了TSharding和Raptor分库分表中间件；

- 服务内部有异步化处理需求，有了Corgi MQ；
- 服务中心内部有分布式缓存需求，所以有了MoguCache等。

所以服务框架是支撑和推动整个网站架构发展的核心和源动力。2015 年底我们启动了 PHP Web 单体应用拆分为一个个 Java Web 业务应用和前后端分离项目，很快购买链路完成了 PHP 体系到 Java 生态的彻底升级。

Tesla 走向真正的服务框架（而不仅仅是一个 RPC 框架）是随着业务的不断发展和业务系统的强烈要求开始的。最早业务系统面临服务发布时不平滑（客户端几秒甚至数 10 秒后才拉到更新的 IP 列表）等问题，以及业务服务蓬勃发展时期强烈的服务治理需求，Tesla 不断进行服务框架的改造升级：

- 通过新的配置中心Metabase的诞生和客户端拉取配置优化、客户端响应服务端连接关闭事件解决连接优化问题；
- 通过对监控预警、动态限流、服务端超时、服务分组等的支持完善了服务治理体系。

实际上业界的众多服务框架也是这

么一步步发展起来的，没有捷径。

中间件在全站的有效落地，是靠中间件团队和业务系统支撑团队密切配合完成的。新的中间件刚刚开发出来，肯定会有各种问题出现，但是中间件也是业务系统的实际需求推动才产生的，所以双方的目标、利益点是相同的。我们会采用项目的方式，每个项目成员都对中间件的使用推广和自身完善负责，共同推动中间件在全站的稳定落地。

### **InfoQ：在中间件研发过程中，开源技术起到怎样的作用？**

**七公：**蘑菇街是拥抱开源技术的。

在我们的技术体系中，很大一部分基础技术组件依赖于开源产品。我们的Tesla服务框架是基于Netty开发的，最早我们用的MQ是RabbitMQ，后来我们才转向自研；我们的Sentry监控平台是基于Grafana；目前我们仍然在广泛使用Kafka来收集全站日志，使用Zookeeper协调分布式系统，使用Hadoop生态来支撑数据平台计算任务，等等。

而自研Tesla、Corgi MQ等中间件的原因是因为这些是支撑我们网站发展的核心产品，我们必须能够完全掌控、有能力做到深度定制，以便快速支持业务发展，而不是成为业务发展的瓶颈和

阻碍；而这些中间件的诞生、持续发展和在全站落地之后，变成了我们技术团队乃至整个蘑菇街的核心竞争力之一。

针对纯技术的中间件 / 产品，我们会选择性的开源。TSharding 是我们最早的分库分表中间件，之前因为小米、有赞等不少技术同仁都有咨询，所以我们开放在了 Github 上，见 <https://github.com/baihui212/tsharding>。

**InfoQ: 如何通过数据分析指导“前瞻性地谋划实施、支撑业务快速发展”？在电商中哪些数据发生怎样的变化会给系统架构带来预警，提示需要改良或优化？能否结合谈谈你们的峰值系统的监控架构和方案？**

**七公：**“前瞻性地谋划实施”可以通过关注业务数据和系统数据的增长变化情况来获得一个合理的度。通过当前业务的增速能推算出来三五年内的业务爆发和数据增长情况，如果有些架构上的不适宜（如过度中心化或存在易攻击点、系统难以扩展或者稳定性难以保障等问题），要尽早规划新的架构形态并快速推进实施，提早解除可能阻碍业务发展的绊脚石。但是也不能过度设计，考虑五年以后、十年以后的情况做设计，可能还未到时候，现有架构早就过时了。

电商数据比较复杂，概括如下：

## 业务数据

目前我会关注 DAU，分平台的 GMV、UV、客单价、支付订单数、支付用户数等。

## 系统数据

主要是容量和系统运行情况，容量包括数据库表空间、磁盘使用情况、服务应用的水位、缓存集群的分片情况等；系统运行状态主要包括 QPS、RT、调用成功率、CPU 占用率、内存使用、IO 等。

只要持续关注这些核心数据，才能敏锐地把控到系统接下来的发展趋势和改造方向。

蘑菇街大促峰值的监控分两部分：一部分是实时业务监控，播报实时 PV、UV、购买订单数、GMV、客单价的情况，这部分是通过实时计算平台的流式计算任务来完成的，延迟在秒级左右；另一部分是实时系统监控，我们是通过异步上报、LogAgent 收集、实时统计分析来保证 10s 左右的延迟就能把系统的实时统计指标呈现在可视化监控平台上。前者给业务方密切关注，后者则是我们做大促保障指挥决策、预案执行的哨兵和自动化告警的主要手段。

**InfoQ: 目前蘑菇街的架构是否已经达到了“该优化的已经优化了”的阶段？您将在 ArchSummit 的分享中提到“25 倍”的优化路径，在您的猜想下，**

## 25 倍是否是优化的极限，您认为短期未来还可以达到多少，如何做？

**七公：**蘑菇街业务每年都保持 3 倍多的迅猛增长，对技术的挑战一直非常大。2015 年我们主要完成了业务应用服务化建设 & 服务架构升级、基础中间件研发落地、前后端分离、稳定性平台建设 & 大促保障常态化等重大技术改造，为网站架构的进一步发展打下了很好的基础。

目前我们还面临诸多问题和挑战，比如前端组件化的建设和新业务的快速交付、蘑菇街 / 美丽说 / 淘世界整个集团各平台业务发展的快速支撑和系统化的稳定性保障 & 质量保障能力；JDK8、异步框架等技术在全站的落地；应用高性能的持续优化、虚拟化 & 私有云的持续建设等等，蘑菇街还远未到“该优化的已经优化了”的阶段。我们要做的就是每行代码都写好、每件事情都做到位，让我们的系统变的越来越强大，同时用更少的人力和机器成本去支持更大体量的业务。

关于优化的极限问题，最终追求的极限不是倍数，是在机器数上。机器堆的多了，自然可以支撑更大的流量，但是我们目前在做的是用更少的机器，支持更多倍的流量。目前我们主要还是进

一步提高虚级化占比的方式（已逼近上限）和单机能力提升（还大有可为）的方式来进行的。

**InfoQ：为什么您认为“根据场景来选用性能优化方案，没有通用方案”，蘑菇街以前是否考虑过“通用方案”的设计？能否结合蘑菇街架构升级案例谈谈体会？**

**七公：**性能优化有很多种手段。

### 硬件

对硬件更新换代、提升硬件能力、调优硬件参数可以提升性能，横向扩展方面加机器也可以提升性能。

### 软件

对数据库 / Web Server 等软件进行配置优化、对 CPU 执行效率做优化、对 IO 效率做优化、对程序 CPU 计算复杂度做优化等等都可以提升性能，甚至仅仅做业务化简也能得到很可观的 RT 的降低。

2015 年初蘑菇街面临非常大的性能问题，只能支撑每秒 400 单左右的交易量，严重不满足业务超高速发展和大促迅猛流量的要求。我们在完成系统拆分 & 服务化 1.0 体系构建后，开始购买链路的专项性能提升项目。这个项目过程中翻越了许多障碍、篱笆，比如 DB 单点写瓶颈是一直压在头上的一座大山。

我们通过自研分库分表组件 TSharding，完成分库分表，最终下单服务支持了写入的无限水平扩展，订单库容量提升到了千亿级规模；营销服务的 RT 不稳定，每轮压测，营销服务都能再进一步，然后遇到下一个问题；我们通过 SQL 优化、缓存 & 预处理、读写分离等手段，大促期间最终计价接口 RT 稳定在 7ms；异步处理非常有必要性，能大大降低服务响应 RT，然后通过自己的方式解决异步化后的分布式事务问题，等等。

我们优化的案例还有很多，优化的过程很类似：找到瓶颈点——优化——压测验证然后再循环。但是最优的方案

往往都是针对该场景和问题定制的，不应该去追求通用方案，也不会有通用方案。

**InfoQ：并没有很多技术人员能有处理海量服务的机会，在从事这方面的工作中您有什么特别的感悟或经验可以和大家分享吗？**

**七公：**我的体会是：只有自己经历过、尝试过的，才能真正成为自己的。如果有意愿深入接触高并发高可用高可扩展服务架构，但是目前还没有机会的，建议尽早换到业务在快速发展的有大流量场景的互联网公司，给自己挑战自我的机会。

主办方 **Geekbang** 极客邦科技

**X-NODE** 创极无限

战略合作伙伴 **拉勾** 专注互联网职业机会 [www.lagou.com](http://www.lagou.com)

12月4日 - 12日 硅谷

# ChinaTech Day

## 中国技术开放日·美国站

# 邀您同行！

30 位技术大牛同行，探访硅谷 10 家创新科技企业

和我们一起去硅谷！





# 深度访谈

## 华为开源数据格式CarbonData项目，实现大数据即席查询秒级响应

作者 杜小芳



华为宣布开源了CarbonData项目，该项目于6月3日通过Apache社区投票，成功进入Apache孵化器。CarbonData是一种低时延查询、存储和计算分离的轻量化文件存储格式。那么相比SQL on Hadoop方案、传统NoSQL或相对ElasticSearch等搜索系统，CarbonData具有什么样的优势呢？CarbonData的技术架构是什么样子的？未来有什么样的规划？我们采访了

CarbonData项目的技术负责人为大家解惑。

**InfoQ：请问CarbonData是什么时候开始进行的项目？为什么现在向Apache孵化器开源呢？开源发展历程和项目目前状态是怎么样的？**

**CarbonData：**CarbonData项目是华为公司从多年数据处理经验和行业理解中逐步积累起来的，2015年我们对系统进行了一次架构重构，使其演化为

HDFS 上的一套通用的列式存储，支持和 Spark 引擎对接后形成一套分布式 OLAP 分析的解决方案。

华为一直是面向电信、金融、IT 企业等用户提供大数据平台解决方案的供应商，从众多客户场景中我们不断提炼数据特征，总结出了一些典型的对大数据分析的诉求，逐步形成了 CarbonData 这个架构。

因为在 IT 领域，只有开源开放，才能最终让更多的客户和合作伙伴的数据连接在一起，产生更大商业价值。开源是为了构建 E2E 生态，CarbonData 是数据存储层技术，要发挥价值，需要与计算层、查询层有效集成在一起，形成完成真正的生态发挥价值。

又因为 Apache 是目前大数据领域最权威的开源组织，其中的 Hadoop，Spark 已成为大数据开源的事实标准，我们也非常认可 Apache 以 Community 驱动技术进步的理念，所以我们选择进入 Apache，与社区一同构建能力，使 CarbonData 融入大数据生态。

目前 CarbonData 开源项目已经在 6 月 3 日通过 Apache 社区投票，成功进入 Apache 孵化器。

相关社区信息如下：Apache CarbonData github 地址：<https://github.com/>

[apache/incubator-carbondata](https://github.com/apache/incubator-carbondata)。

欢迎大家参与到 Apache CarbonData 社区：<https://github.com/apache/incubator-carbondata/blob/master/docs/How-to-contribute-to-Apache-CarbonData.md>。

**InfoQ：请问是什么原因或机遇促使您们产生做 CarbonData 这个项目的想法的？之前的项目中遇到什么样的困难？**

**CarbonData：**我们一直面临着很多高性能数据分析诉求，在传统的做法里，一般是使用数据库加 BI 工具实现报表、DashBoard 和交互式查询等业务，但随着企业数据日益增大，业务驱动的分析灵活性要求逐渐增大，也有部分客户希望有除 SQL 外更强大的分析功能，所以传统的方式渐渐满足不了客户需求，让我们产生了做 CarbonData 这个项目的想法。

需求一般来源于几方面。

第一，在部署上，区别于以往的单机系统，企业客户希望有一套分布式方案来应对日益增多的数据，随时可以通过增加通用服务器的方式 scale out 横向扩展。

第二，在业务功能上，很多企业的业务都处在从传统数据库逐渐转移到大



数据平台的迁移过程中，这就要求大数据平台要有较高兼容老业务的能力，这里面主要包含的是对完整的标准 SQL 支持，以及多种分析场景的支持。同时为了节约成本，企业希望“一份数据支持多种使用场景”，例如大规模扫描和计算的批处理场景，OLAP 多维交互式分析场景，明细数据即席查询，主键低时延点查，以及对实时数据的实时查询等场景，都希望平台能给予支持，且达到秒级查询响应。

第三，在易用性上，企业客户以往使用 BI 工具，业务分析的 OLAP 模型是需要 BI 工具中建立的，这就会导致有的场景下数据模型的灵活性和分析手段受到限制，而在大数据时代，大数据开源领域已经形成了一个生态系统，社区随时都在进步，经常会冒出一些新型的分析工具，所以企业客户都希望能跟随社区不断改进自己的系统，在自己的数据里快速用上新型的分析工具，得到更大的商业价值。

要同时达到上诉要求，无疑对大数据平台是一个很大的挑战。为了满足这些要求，我们开始不断在实际项目中积累经验，也尝试了很多不同的解决方案，但都没有发现能用一套方案解决所有问题。

大家首先会想到的是，在涉及到低时延查询的分布式存储中，一般常用的是 KV 型 NoSQL 数据库（如 HBase，Cassandra），可以解决主键低时延查询的问题，但如果业务的查询模式稍作改变，例如对多维度灵活组合的查询，就会使点查变为全表扫描，使性能急剧下降。有的场景下，这时可以通过加入二级索引来缓解该问题，但这又带来了二级索引的维护和同步等管理问题，所以 KV 型存储并不是解决企业问题的通用方案。

那么，如果要解决通用的多维查询问题，有时我们会想到用多维时序数据库的方案（如 LinkedIn Pinot），他们的特点是数据都以时间序列的方式进入系统并经过数据预聚合和建立索引，因为是预计算，所以应对多维查询时非常快，数据也非常及时，同时具备多维分析和实时处理的优点，在性能监控、实时指标分析的场景里应用较多。但它在支持的查询类型上也有一定限制，因为做了数据预计算，所以这种架构一般无法应对明细数据查询，以及不支持 Join 多表关联分析，这无疑给企业使用场景带来了一定的限制。

另外一类是搜索系统（如 Apache Solr，ElasticSearch），搜索系统可

以做多维汇总也可以查询明细数据，它也具备基于倒排索引的快速布尔查询，并发也较高，似乎正是我们希望寻找的方案。但在实际应用中我们发现两个问题：一是由于搜索系统一般是针对非结构化数据而设计的，系统的数据膨胀率一般都比较低，在企业关系型数据模型下数据存储不够紧凑，造成数据量较大，二是搜索系统的数据组织方式和计算引擎密切相关，这就导致了数据入库后只能用相应的搜索引擎处理，这又一定程度打破了企业客户希望应用多种社区分析工具的初衷，所以搜索系统也有他自己的适用场景。

最后一类系统，就是目前社区里大量涌现的 SQL on Hadoop 方案，以 Hive, SparkSQL, Flink 为代表，这类系统的特点是计算和存储相分离，针对存储在 HDFS 上的文件提供标准 SQL 功能，他们在部署性和易用性上可以满足企业客户需求，业务场景上也能覆盖扫描，汇聚，详单等各类场景，可见可以将他们视为一类通用的解决方案。为了提高性能，Spark, Flink 等开源项目通过不断优化自身架构提升计算性能，但提升重点都放在计算引擎和 SQL 优化器的增强上，在存储和数据组织上改进并不是重点。

所以，可以看出当前的很多大数据系统虽然都能支持各类查询场景，但他们都是偏向某一类场景设计的，在不是其目标场景的情况下要么不支持要么退化为全表扫描，所以导致企业为了应对批处理，多维分析，明细数据查询等场景，客户常常需要通过复制多份数据，每种场景要维护一套数据。

CarbonData 的设计初衷正是为了打破这种限制，做到只保存一份数据，最优化地支撑多种使用场景。

### **InfoQ: 能否具体谈谈 CarbonData 的技术架构？有何特征和优势呢？**

**CarbonData:** 整个大数据时代的开启，可以说是源自于 Google 的 MapReduce 论文，他引发了 Hadoop 开源项目以及后续一系列的生态发展。他的“伟大”之处在于计算和存储解耦的架构，使企业的部分业务（主要是批处理）从传统的垂直方案中解放出来，计算和存储可以按需扩展极大提升了业务发展的敏捷性，让众多企业普及了这一计算模式，从中受益。

虽然 MapReduce 开启了大数据时代，但它是通过纯粹的暴力扫描 + 分布式计算来提升批处理性能，所以并不能解决客户对所有查询场景的低时延查询要求。

在目前的生态中，最接近于客户要求的其实是搜索引擎类方案。通过良好的数据组织和索引，搜索引擎能提供多种快速的查询功能，但偏偏搜索引擎的存储层又和计算引擎是紧耦合的，并不符合企业对“一份数据，多种场景”的期望。

这给了我们启发，我们何不为通用计算引擎打造更一个高效的数据组织来满足客户需求呢，做到既利用计算和存储解耦架构又能提供高性能查询。抱着这个想法，我们启动了 CarbonData 项目。针对更多的业务，使计算和存储相分离，这也成了 CarbonData 的架构设计理念。

确立了这个理念后，我们很自然地选择了基于 HDFS+ 通用计算引擎的架构，因为这个架构可以很好地提供 Scale out 能力。下一步我们问自己这个架构里还缺什么？这个架构中，HDFS 提供文件的复制和读写能力，计算引擎负责读取文件和分布式计算，分工很明确，可以说他们分别定位于解决存储管理和计算的问题。但不难看出，为了适应更多场景，HDFS 做了很大的“牺牲”，它牺牲了对文件内容的理解，正是由于放弃了对文件内容的理解，导致计算只能通过全扫描的方式来进行，可以说最终导致的是存储和计算都无法很好的利用数

据特征来做优化。

所以针对这个问题，我们把 CarbonData 的发力重点放在对数据组织的优化上，通过数据组织最终是要提升 IO 性能和计算性能。为此，CarbonData 做了如下工作。

### CarbonData 基础特性

- 多维数据聚集：在入库时对数据按多个维度进行重新组织，使数据在“多维空间上更内聚”，在存储上获得更好的压缩率，在计算上获得更好的数据过滤效率。
- 带索引的列存文件结构：首先，CarbonData 为多类场景设计了多个级别的索引，并融入了一些搜索的特性，有跨文件的多维索引，文件内的多维索引，每列的 minmax 索引，以及列内的倒排索引等。其次，为了适应 HDFS 的存储特点，CarbonData 的索引和数据文件存放在一起，一部分索引本身就是数据，另一部分索引存放在文件的元数据结构中，他们都能随 HDFS 提供本地化的访问能力。
- 列组：整体上，CarbonData 是一种列存结构，但相对于行存来说，列存结构在应对明细数据

查询时会有数据还原代价高的问题，所以为了提升明显数据查询性能，CarbonData支持列组的存储方式，用户可以把某些不常作为过滤条件但又需要作为结果集返回的字段作为列组来存储，经过CarbonData编码后会将这些字段使用行存的方式来存储以提升查询性能。

- 数据类型：目前CarbonData支持所有数据库的常用基本类型，以及Array，Struct复杂嵌套类型。同时社区也有人提出支持Map数据类型，我们计划未来添加Map数据类型。
- 压缩：目前CarbonData支持Snappy压缩，压缩是针对每列分别进行的，因为列存的特点使得压缩非常高效。数据压缩率基于应用场景不同一般在2到8之间。
- Hadoop集成：通过支持InputFormat/OutputFormat接口，CarbonData可以利用Hadoop的分布式优点，也能在所有以Hadoop为基础的生态系统中使用。

## CarbonData 高级特性

可计算的编码方式：除了常见的Delta，RLE，Dictionary，BitPacking等编码方式外，CarbonData还支持将多列进行联合编码，以及应用了全局字典编码来实现免解码的计算，计算框架可以直接使用经过编码的数据来做聚合，排序等计算，这对需要大量shuffle的查询来说性能提升非常明显。

与计算引擎联合优化：为了高效利用CarbonData经过优化后的数据组织，CarbonData提供了有针对性的优化策略，目前CarbonData社区首先做了和Spark的深度集成，其中基于SparkSQL框架增强了过滤下压，延迟物化，增量入库等特性，同时支持所有DataFrame API。相信未来通过社区的努力，会有更多的计算框架与CarbonData集成，发挥数据组织的价值。

目前这些特性都已经合入Apache CarbonData 主干，欢迎大家使用。

**InfoQ：在哪些场景推荐使用呢？  
性能测试结果如何？有没有应用案例，  
目前在国内的使用情况和用户规模？**

**CarbonData：**推荐场景：希望一

份存储同时满足快速扫描，多维分析，明细数据查询的场景。在华为的客户使用案例中，对比业界已有的列存方案，CarbonData 可以带来 5~30 倍性能提升。

性能测试数据及应用案例等更多信息，请关注微信公众号 ApacheCarbonData，及社区 <https://github.com/apache/incubator-carbondata>。

**InfoQ: CarbonData 能和当前正火的 Spark 完美结合吗？还能兼容哪些主流框架呢？**

**CarbonData:** 目前 CarbonData 已与 Spark 做了深度集成，具体见上述高级特性。

**InfoQ: 您的项目在未来有什么样的发展规划？还会增加什么功能吗？如何保证开源之后的项目的持续维护工作呢？**

**CarbonData:** 接下来社区重点工作是，提升系统易用性、完善生态集成（如：与 Flink, Kafka 等集成，实现数据实时导入 CarbonData）。

CarbonData 开源的第一个月，就有几百个 commits 提交，和 20 多个贡献者参与，所以后续这个项目会持续的活

跃。10 多个核心贡献者也将会持续参与社区建设。

**InfoQ: 在 CarbonData 设计研发并进入 Apache 孵化器的过程中，经历了哪些阶段，经历过的最大困难是什么？有什么样的感受或经验可以和大家分享的吗？**

**CarbonData:** CarbonData 团队大多数人都有参与 Apache Hadoop、Spark 等社区开发的经验，我们对社区流程和工作方式都很熟悉。最大的困难是进入孵化器阶段，去说服 Apache 社区接纳大数据生态新的高性能数据格式 CarbonData。我们通过 5 月份在美国奥斯丁的开源盛会 OSCON 上，做 CarbonData 技术主题演讲和现场 DEMO 演示，展示了 CarbonData 优秀的架构和良好的性能效果。

**InfoQ: 您们是一个团队吗？如何保证您们团队的优秀成长？**

**CarbonData:** CarbonData 团队是一个全球化的（工程师来自中国、美国、印度）团队，这种全球化工作模式的经验积累，让我们能快速的适应 Apache 开源社区工作模式。

# 专访丁奇：

## 阿里云即将开源AliSQL，超大并发、针对秒杀优化

作者 韩婷

**丁奇**，阿里云关系数据库服务内核开发和运维团队负责人，活跃的 MySQL 社区贡献者。专注于数据存储系统、MySQL 源码研究和改进、MySQL 性能优化和功能改进。

AliSQL 是基于 MySQL 官方版本的一个分支，由阿里云数据库团队维护，目前也应用于阿里巴巴集团业务以及阿里云数据库服务。该版本在社区版的基础上做了大量的性能与功能的优化改进。尤其适合电商、云计算以及金融等行业环境。该版本性能优于社区版 MySQL 70% 左右，可帮助中小企业和开发者提

升数据运营能力。

阿里云数据库资深专家丁奇介绍，AliSQL 版本在强度和广度上都经历了极大的考验。最新的 AliSQL 版本不仅从其他开源分支比如：Percona，MariaDB，WebScaleSQL 等社区汲取精华，也沉淀了阿里巴巴多年在 MySQL 领域的经验和解决方案。AliSQL 增加更多监控指标，





并针对电商秒杀、物联网大数据压缩、金融数据安全等场景提供个性化的解决方案。

丁奇表示，“在通用基准测试场景下，AliSQL 版本比 MySQL 官方版本有着 70% 的性能提升。在秒杀场景下，性能提升 100 倍”。这样的性能提升数据是如何得出的？即将开源的 AliSQL 又将为开发者带来哪些功能？针对于不同行业，AliSQL 做了哪些工作呢？

近日，InfoQ 就相关问题采访了丁奇。

**InfoQ: AliSQL 是基于 MySQL 官方版本的一个分支，能否简单介绍下 AliSQL 的历史？AliSQL 版本在强度和广度上都经历了哪些考验？**

丁奇：大概在 2009 年，阿里巴巴

集团开始大规模的使用 MySQL 数据库来持久化业务数据。随着集团业务的高速发展，官方的 MySQL 版本遇到了不小的挑战，包括性能、部署、功能、成本等方面。

伴随着业务的驱动和对源代码的熟悉，集团开始尝试在 MySQL 官方的开源版本上进行修改，这就形成了 AliSQL 的雏形。

电商业务的高速发展，对 MySQL 的性能提出了更高的要求。出于节省成本，AliSQL 持续进行了性能优化，同时，多核 CPU 和 SSD 等新硬件的采用，也反过来促进 AliSQL 能够尽可能利用硬件的红利。

双 11 大促不断刷新记录，数据库的稳定性也变得越来越重要，AliSQL 开



始定制基于限流、线程池、秒杀等功能的 patch，提升 AliSQL 的稳定性。

针对小微金融业务对数据保护的高要求，AliSQL 定制了适合金融业务的数据保护方案，例如金融云上使用的双通道日志高可靠方案。

从阿里云 RDS 上线服务开始，阿里云数据库团队就遇到了前所未有的挑战，不同的行业用户，不同的使用习惯和要求，AliSQL 也迎来了发展最为迅速的时期，影响力也越来越大。

所以，AliSQL 的版本，是伴随着业务的发展，一起成长起来的，经历过双 11 大促这样大压力的考验，同时也经历了阿里云各行各业用户差异化的需求。可以说是身经百战。

**InfoQ：阿里云是开源组织 WebScaleSQL 的第五位成员，与 Facebook、Google、Twitter 和 LinkedIn 团队共同研发 WebScaleSQL，同时，阿里巴巴还拥有 OceanBase 自研数据库。能否请介绍下 AliSQL、WebScaleSQL、OceanBase 三者的关系和各自的特点？**

**丁奇：**OceanBase 是 Alibaba 集团自研的分布式数据库，经历了集团业务的洗礼，具有通用性，高扩展能力。

WebScaleSQL 是由这五家公司发起

的基于 MySQL 官方的一个分支，旨在解决大家在互联网业务上遇到的问题，是五个成员公司将各自足够通用的功能提交到一起的集合，每家公司的研发同学都可以提交代码。实际上每个公司自己生产环境使用的是自己维护的一个分支，因为每个公司都有自己定制化的需求。

AliSQL 同样基于 MySQL 官方版本，汲取了官方和社区的技术红利，具有很高的性能和稳定性，并适应不同行业的特点进行了定制。AliSQL 的改进方向主要集中在安全性、稳定性、性能、新功能等方面。

AliSQL 是经过几年的生产环境、几万个用户实例的实际业务锤炼的。

AliSQL 的一些定制化功能都是为了解决 DBA 维护、业务使用中碰到的实际问题。比如 5.5 以上的版本由于有 metadata lock，DBA 对表加字段等操作可能导致阻塞查询，进而导致整库不可服务。我们新增 alter ..wait N .. 方法，保证了操作的安全性。再比如通过提供 set rds\_reset\_connection 这样的语句，解决了长连接占用资源和短连接性能问题的矛盾。

**InfoQ：AliSQL 在相关报道中提到，“在通用基准测试场景下，AliSQL 版**

**本比 MySQL 官方版本有着 70% 的性能提升。在秒杀场景下，性能提升 100 倍。”能否详细讲解该数据是如何得出的？**

**丁奇：**通用基准的测试，我们是采用 sysbench 进行的测试，也是公开的标准测试方法，AliSQL 在吞吐能力上，比 MySQL 官方大概 70% 的性能提升。

秒杀场景是一个比较特殊的场景，AliSQL 有专门的定制 patch 针对这种场景的优化，如果没有限流和排队，大并发的请求下，系统很容易产生雪崩效应，导致吞吐量急剧下降，而非线性关系。所以，秒杀场景下，在不可预知的业务请求量的时候，类似减库存这样的场景，性能下跌非常厉害，而 AliSQL 的秒杀解决方案能够保证这类场景维持高性能。

**InfoQ：电商行业的环境特点是什么样的？AliSQL 在电商行业环境下的优势有哪些？或者说，为了更好地符合电商行业的需求，AliSQL 做了哪些性能与功能的优化工作？**

**丁奇：**电商行业的环境，其实提供了一个非常综合的场景，在扩展性、稳定性、性能等方面对数据库都提出了非常高的要求，AliSQL 就是顺应着这样的要求进行的定制版本。

比如应对大量应用集群的线程池功能，秒杀场景的排队功能，以及 SQL 的限流功能。

比如大写入量备库延迟，AliSQL 提供的基于表的并行复制功能，又比如结构化数据的压缩功能等。

**InfoQ：电商秒杀场景有什么特点，为数据库带来的挑战有哪些？AliSQL 针对该场景优化了哪些性能与功能？已经应用过的实际场景有哪些？实际效果如何？**

**丁奇：**电商的秒杀场景，其实就是减库存，对数据库而言，就是对一条记录的更新，因为事务的特点，单条记录的更新必须串行完成，但秒杀的特点，就是在某个时刻，大量的并发进行减库存，这就造成了大量的线程因获取不到锁而处在死锁检测状态，消耗了大量的 CPU 资源，最终导致系统无法响应，而引起雪崩效应。

AliSQL 针对这样的场景，提供了排队和限流的功能，经过了双 11 零点时刻高并发请求的考验，保持了系统的稳定性和持续吞吐能力。

电商业务高峰有两个对数据库挑战比较大的场景。

## 1. 超大并发

MySQL 能够支持的并发活跃连接数

“**阿里巴巴是支持开源的，基于开源的协议，有着回馈社区的初衷，并怀着促进社区发展的良好愿景。**”

是有上限的，理想情况下是大约（CPU 核心数  $\times 2$ ）个活跃连接数，当活跃连接数远超这个值时，性能会急剧下降，导致整个业务不可用。AliSQL 有水位控制，超过一定阈值的活跃连接数，当我们判断到当前压力超过数据库的处理能力时，会主动放弃后到的请求，这样保证数据库还能保持很高的能够正常响应的吞吐量。

## 2. 秒杀场景

在秒杀场景里面有一个减库存的问题。大量用户同时抢购同一个商品的时候，需要同时更新商品库存，这时候 InnoDB 的行锁加上死锁检测机制会导致数据库 CPU 短时间内被占满，导致整库几乎无法响应。

在 AliSQL 我们专门针对秒杀的方案，保证在大量线程同时减库存时

仍能保持很高的 TPS。除了阿里自己的秒杀业务，这个功能同样适用于抢红包这样的业务，已经在 2015、2016 年春节经过大量的业务验证。

**InfoQ：在个性化方面，AliSQL 针对云计算和金融行业做了哪些优化工作？应用过的实际场景有哪些？实际效果如何？AliSQL 在物联网大数据压缩、金融数据安全等场景又提供了哪些个性化功能？**

**丁奇：**在云计算的环境下，用户的使用场景和方式都千差万别，为了适应不同的环境，AliSQL 定制了很多个性化的功能，比如，为了保障在线业务的平稳，针对用户的分析型的 SQL，AliSQL 提供了资源使用限流、全表扫描 buffer pool 不缓存的特性，用户可以通过设置环境变量或者使用 hint 来方便的使用

这些功能，又比如为了加快大表的扫描，提供了逻辑预读的功能，

这些特性，用户在不同的场景下可以自由选择。

除了这些，AliSQL 在公有云上针对不同行业定制了很多功能。比如：

- 游戏行业

我们在 proxy 这一层进行了 AliSQL 的桥梁认证，提供防闪断功能。

- 物联网行业

AliSQL 集成了 TokuDB 引擎，提供高压缩比和大吞吐写能力。

- 金融行业

AliSQL 定制了多通道的半同步策略，以及一主两备的三机房零数据丢失的数据保护级别。

**InfoQ：针对不同的应用场景，AliSQL 增加了哪些监控指标？**

**丁奇：**AliSQL 增加了很多监控指标，以帮助用户或者 DBA 更了解自己的数据库，比如，AliSQL 针对四个不同的维度的统计：

1. SQL 维度。增加每个 SQL 执行的开销统计，除了响应时间，锁等待等，还包括逻辑读，物理读，临时空间使用等。
2. 对象维度。增加了每个表的 DML 次数，索引的使用情况，帮助用户

理解业务和索引使用效率。

3. 事务维度。增加了每个事务持续的时间，和操作的对象。帮助用户定位问题。
4. 线程维度。增加了线程的内存使用统计。

**InfoQ：阿里巴巴一直在推进开源工作，您认为做好开源工作，公司应该从哪些方面出发？目前，AliSQL 的开源计划是怎样的？将为开发者具体怎样的帮助？**

**丁奇：**我觉得，公司首先是支持开源的，基于开源的协议，有着回馈社区的初衷，并怀着促进社区发展的良好愿景。

AliSQL 开源会保持着一个好的节奏，持续的高质量回馈社区。具体在 9 月中旬放出 binary，10 月份会放出源代码的第一个稳定版本。

开发者可以自由下载使用，并在平台上进行反馈或者提出建议，后续 AliSQL 也会定期组织论坛，邀请开发者参与进行讨论。

我们希望 AliSQL 能够形成一个活跃的社区，开发者能够从中受益，也能够提出需求和改进建议，促进分支持续发展。

# 百度贴吧王伟冰：

## 跳出历史技术栈，谨慎寻找最佳优化思路

作者 李东辉

**王伟冰**，2010 年加入百度，参与设计开发百度 Memcached、Redis、通用反作弊等通用服务，服务百度 50+ 产品线，日 pv 上万亿。负责百度 LNMP 基础设施(Nginx、PHP、HHVM、RPC 中间件、扩展、基础库等) 开发维护；负责百度 OXP 私有云平台建设；负责百度贴吧性能优化；优化 PC、移动、Native 客户端的用户访问速度；建立了全面的性能评估、分析、监控体系等工作。

从 2003 年诞生到现在，基于兴趣聚合的百度贴吧已经走过了 13 年，随着用户、流量的不断增长，业务的不断复杂化，研发团队的规模不断扩张，百度贴吧这 13 年经历了 LAMP 化、服务化、私有云化等阶段。然而面对每一次架构调整，贴吧是如何在无数优化方向中选择最适合现状同时又兼容未来的技术方案？其中贴吧团队又需要经过多少调研多少试验才能进行判断与落实？现在我们就来采访王伟冰老师，让我们提前预览每一次架构变动背后的细节与故事吧。



**InfoQ：您从 2010 年开始加入百度至今已经 6 年了，能否谈谈这些年您经历了哪些团队变化和成长，这些年给您最大的收获和教训是什么？**

**王伟冰：**前四年我一直在社区基础技术 - 架构团队，做过各种通用服务和通用组件，如 Memcached、Redis、Antispam、HHVM 等等，服务于百度内部的各种业务，包括知识、贴吧、移动云等。近两年也参与了贴吧业务，做过性能优化、业务监控等。

基础技术的视角和业务的视角是不一样的，两方面都做过之后看问题会更全面。比如做基础技术的时候会从业务的角度去想，业务为什么需要这样的

技术，为什么会用你提供的东西；而做业务的时候也会想，这个东西如何抽象，做成更通用的技术让更多人可以使用。

**InfoQ：您曾参与开发设计百度 Memcached、Redis 等数据缓存技术方案，能否谈谈它们各自的业务范围以及为什么选用了它们？能否简单介绍百度的存储架构设计经历了怎样的迭代与发展？**

**王伟冰：**做 Memcached 是 2011 年，当时公司内不少业务在用 Memcached 来做缓存，但都各自运维，在集群化方案、高可用方案、一致性方案上都有一些缺陷，所以我们就做了一个基于 Memcached 的通用缓存服务，并提供封



装好的客户端，用户不需要再关心分布式、高可用上的细节，并提供延迟删除的功能来解决 Cache 与 DB 数据不一致的问题。

当时我们也有高写入性能、持久化存储的需求，比如浏览计数、最近访问列表，那时 Redis 还没有火，我们基于 Berkeley DB 做了一套小数据 kv、klist 存储服务，解决了当时的业务问题。到了 2012 年，我们有了一些更复杂的需求，比如列表 top 排序、过期自动失效，这时我们发现 Redis 这个东西，非常适合做这个事情，而且性能很高，所以开始做 Redis 存储服务。

百度的存储架构，这个 Topic 太大了，比如网页库存储就是一套非常复杂的体系。具体到社区业务而言，主要是经历了 3 个阶段：08 年以前，是专有 C 模块存储；08-12 年，逐步 LAMP 化，Mysql+Cache 成为主流；12 年到今，则是 Mysql+Nosql 并存，Nosql 除了 Memcached/Redis，还有各种百度内部实现的 kv、klist、table 系统，满足各种不同的业务需求。

**InfoQ：你们逐渐从 LAMP 架构转移至 OXP 私有云架构，能否简单介绍开发 OXP 的初衷和背景分别是什么？**  
**OXP 是如何统筹兼顾开发框架、测试环**

**境、运维平台、监控系统等模块以及各模块之间是如何联系如何协同工作的？**

**王伟冰：**LAMP 化、服务化、私有云化是贴吧架构的 3 个重要的变革点，LAMP 化是提升单个模块开发效率，服务化是提升大团队协作开发效率，而私有云化则是提升运维效率。

在 LAMP 化的过程，我们抽取出一套通用的开发框架和平台，称为 ODP(Online Develop Platform)，用这套框架业务可以快速地开发新应用。但是随着业务规模越来越复杂，机器规模越来越大，测试、运维方面的问题变得越来越突出，开发人员更多的时间都花在搭建测试环境、上线、处理线上问题。我们意识到要提升全流程的效率，而不仅仅是开发阶段的效率。因此我们做了全流程的解决方案，包括开发 (Develop)、测试 (Testing)、运维 (Operation)、监控与问题定位 (Monitor)，这就是 OXP 私有云。

在 OXP 私有云中，我们通过定义一系列的规范，比如目录规范、日志规范、RPC 规范，把整个流程串起来。只要用我们的开发框架进行开发，应用就是符合规范的，我们就可以按照这个规范去运维、去构建测试环境、去采集日志进行监控分析。规范就像一个协议，每个



服务、每个平台都遵守这个协议，那么整个流程就可以很好地串起来。

在迁移私有云的过程中，难点在于如此庞大的业务如何平滑地进行迁移。每一次架构的大改动，都需要花费数月的时间。在这个期间，新老系统是并存的，我们设计了一些方案，保证一个业务可以同时存在新系统和老系统，代码自动同步，流量可以自由切换，然后再逐个业务去切换。

**InfoQ：服务端上，贴吧分别在 Native 端和 WebApp 上的优化思路有什么不同，在各端上贴吧应该分别侧重哪些环节和服务？**

**王伟冰：**Native 和 WebApp 存在很大的不同，Native 的行为更可控，我们在网络协议上做了很多优化。WebApp 的优势在于渲染上比较灵活，所以我们做在延迟加载、分段加载等方面做更多功夫。

Native 和 WebApp 的融合也是近年来的一个趋势。我们也在尝试多个思路，比如用配置来控制 Native 的展现、用脚本来控制 Native 展现（如 React Native）、嵌入浏览器内核的 Hybrid 模式，这些方法都各有各的优缺点：

- 配置方式的性能最好，但灵活性差；

- 浏览器方式的灵活性最好，但性能差；
- 脚本方式兼顾了性能和灵活性，但实现成本又较高，更容易出问题。

目前来说还没有看到哪一种有特别明显的优势，所以目前是根据不同业务的需求场景来选择合适的技术，比如产品核心页面更重性能，可以用配置方式；运营类页面更重灵活性，可以用浏览器方式；脚本方式作为一种比较新的技术，可以在新业务上进行尝试。

**InfoQ：您曾负责百度贴吧性能优化，节约过半服务器，能否具体谈谈该阶段的优化背景，你们团队在各环节做了哪些优化以达到这个效果？在贴吧性能优化上有没有长期的指导思想（优化思路）？**

**王伟冰：**14 年初的时候，贴吧流量涨得很厉害，服务器规模大涨，所以决定做性能优化。最主要是做了两件事情，RPC 中间件优化和迁移 HHVM，这个优化效果是贴吧历史上优化最好的，节省了过半的前端服务器。性能优化主要有两个思路，一是尝试引入新技术（如 HHVM），二是分析业务的性能瓶颈并做针对性的优化，这两个方案往往是最有效的。

新技术的引入是需要比较谨慎的，选择 HHVM 我们也是观察了较长时间，在内部做了很多试点，发现 HHVM 性能确实很高，兼容性、稳定性也还不错，同时社区很活跃，所以才决定全面迁移。除了主动地做性能优化，更重要的是防止退化。我们现在设计了一些监控算法，能够实时地发现性能的变化。当发现性能变化时，我们会对比分析性能的构成，看看哪部分的模块或者哪部分的函数耗时发生了变化，同时我们也会查找当时的变更事件，看看是否是有相关的上线变更操作，导致性能发生了变化。

**InfoQ: 贴吧已经是 13 岁的“老”产品了，您和您的同事是否遇到过不敢处理的“似是而非”的代码？遇到这种情况应该怎么处理？能否分享面对具有大量历史代码的产品的处理经验？**

**王伟冰:** 老代码肯定是存在的。不过由于贴吧历史上经历了多次大规模的架构改造，大部分代码都被重写了，剩下小部分，也是一些很边缘、不重要的功能，平时并不怎么会接触到。

在性能优化的过程中，我们是尽量避免去改业务代码的。即使要改，也会尽量地兼容以前的逻辑，不管这些代码是新的还是老的。当做大的重构项目的

时候，我们会去梳理所有的业务逻辑，对于不再有用的代码，和产品确认之后，就会去掉。

**InfoQ: 从 13 年开始贴吧已经面临 100 亿流量的规模了，能否谈谈现在移动化 + 社交背景下，贴吧架构现在面临了哪些瓶颈、遇到了哪些新的挑战并且如何应对？**

**王伟冰:** 性能优化是无止境的，一方面业务还在变得更复杂，流量还在涨；另一方面，比较好做的优化我们都已经做了，未来的优化将会更难，这对我们是很大的挑战。这时候需要跳出原来的技术栈，从更上层或者更低层的技术栈来看优化。比如从上层角度看，业务需求本身是可以优化的，例如有些需求不一定合理，有些功能不一定有用，把这些干掉性能就上去了；从底层角度看，资源层还有较大的优化空间，通过资源的合理混布、智能调度可以让资源更加节省。

除了性能之外，研发效率是更大的挑战。从业务上看，贴吧的产品形态正在变得越来越多元化，对研发效率的要求也越来越高，我们也在不断地完善私有云，以支撑业务更快地迭代。

www.egonetworks.org



在微信关注 EGO

## 关于 EGO

EGO 是极客邦科技旗下高端技术人聚集和交流的组织，以 CTO、技术 VP 等技术领导者为服务对象，旨在组建全球最具影响力的技术领导者社交网络，线上线下相结合，推动杰出的技术领导者学习和成长。

EGO 采用实名付费会员制，每一位申请加入的会员需经过严格审核和面试，保证信息的真实性以及会员质量，让处于同等级别的技术管理者在平等和相互信任的环境中分享交流。EGO 目前已建立北京、上海、广州、深圳、杭州五大分会，也将继续在互联网发展集聚的城市开拓分会，覆盖国内大部分城市地区。

## 会员收益

### 学习交流

私密独享线下小组交流，顶尖技术专家、商业领袖分享经验，跨领域、跨地区闭门会议，软技能提升。

### 拓展人脉

EGO 年度盛会，企业间互访，认识国内外志同道合的技术同行。

### 解决问题

遇到问题通过 EGO 平台第一时间找到答案，成为技术领导者身边的智囊团。

### 获取信息

获取前沿行业资讯，独家原创专家观点，社会热点解析。

### 会员福利

优先参与 InfoQ 中国主办的各类技术会议，享受会员购票优惠政策，连接众多会员企业资源，打造会员企业间专享资源生态圈。



## 关于 StuQ

在职技术人的一个学习方式。

StuQ 是极客邦科技旗下职业教育品牌，帮助 IT 从业者和技术团队提高研发水平，为 0-5 年的 IT 从业者提供线上和线下技术学习服务。

StuQ 从 2016 年 1 月份开始正式对外提供学习服务，目前已经有来自 7 个技术领域的 173 名一线技术专家作为讲师在 StuQ 分享了自己的实战技术经验，有超过 6 万多名学员在 StuQ 提高了自己的职业技能，有 267 家企业通过 StuQ 提升了技术团队的研发能力。

## 6 大职业技术角色课程



运维工程师



前端工程师



大数据工程师



Android 工程师



iOS 工程师



PHP 工程师

## 个人学习服务

为初入职场的技术人提供学习服务，通过企业一线技术专家实战经验传授，全面提升个人的研发能力。

### 众筹小班课

直播辅导教学，1-3 位领域专家与 30 位学员团结在一起，连续 7 周，只为一个共同目标，学会一个职业技能。全程使用 StuQ 视频直播和在线协作学习工具。

### 会员学习计划

会员制学习，基于小组的同行交流和学习服务。不但可以与同行在一个学习群交流，还可以两周一次、长期参加视频直播技术案例分享。

## 企业学习服务

为企业技术团队提供学习服务，通过资深技术专家带来的实战课程和专业学习管理工具，帮助企业的整个技术团队提升研发能力。

### 企业内训

针对企业内部研发团队学习计划和技能需要，通过一线技术专家线上沟通诊断，为企业定制专业的技术培训服务，提供定制化、多样化、职能化的研发体系课程，帮助企业解决软件研发过程中遇到的问题和困难，促进软件企业研发团队迅速成长。

### 工作坊

针对不同企业软件研发团队技术人员开展的线下精品小班课，携手知名一线资深技术专家，为技术人成长提供必修精选课程，课程设计系统全面、注重实践分享，帮助技术人员提升职业技能。

# 十八大专题 场场精彩

日新月异的移动架构

机器学习实战

社交网络与视频直播

微服务

容器

运维创造价值的时代

大数据处理及系统架构

...

北京 · 国际会议中心

2016年12月2日-12月3日

扫码查看更多精彩详情



想了解更多？  
进入官网查看精彩内容

[bj2016.archsummit.com](http://bj2016.archsummit.com)



国内最好的原创技术社区，一线互联网公司核心技术人员提供优质内容。订阅 InfoQ，看全球互联网技术最佳实践。做技术的不会没听过 QCon，不会不知道 InfoQ 吧？——冯大辉  
从事技术工作，或有兴趣了解 IT 技术行业的朋友，都值得订阅。——曹政



关注「InfoQ」回复“大咖说”，看大咖聊职业发展、行业洞察



### 聊聊架构

以架构之“道”为基础，呈现更多的务实落地的架构内容。

关注「聊聊架构」  
和百位架构师共聊架构



### 细说云计算

探讨云计算的一切，关注云平台架构、网络、存储与分发。这里有干货，也有闲聊。

关注「细说云计算」  
回复“群分享”，  
看云计算实践干货分享文章



### 大数据杂谈

专注大数据和机器学习，分享前沿技术，交流深度思考。

关注「大数据杂谈」  
回复“微课堂”，看历届  
高质量技术分享文章汇总



### 前端之巅

紧跟前端发展，共享一线技术，不断学习进步，攀登前端之巅。

关注「前端之巅」  
回复“京东”，看京东  
如何做网站前端监控



### 移动开发前线

关注移动开发领域最前沿和第一线开发技术，打造技术分享型社群。

关注「移动开发前线」  
回复“群分享”，看移动  
开发实践干货文章



### 高效开发运维

常规运维、亦或是崛起的DevOps，探讨如何IT交付实现价值。

关注「高效开发运维」  
回复“DevOps”，四篇精品  
文章领悟DevOps





# 博文视点

## IT 出版旗舰品牌

创新图书服务体制 · 打造精品图书品牌



大型网站性能监测  
分析与优化



轻量级微服务架构  
(上册)



SRE: Google运维解密



解析深度学习：语音识别实践



Vue.js权威指南



关注送书

回复：博文视点送书活动  
大会期间每天抽取五名  
可至博文视点展位兑换

# 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），它一直是我們了解业界研发趋势，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有潜质成为国内最专业、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