

T O P T E C H T E A M

# 中国顶尖 CHINA 顶尖

2017年／第十季

技术团队  
访谈录

「Qcon特刊」



扫一扫，了解更多

# ArchSummit

全 球 架 构 师 峰 会 2017

12.8-9 北京 · 国际会议中心



国内外优秀架构设计要领及实践秘诀

From : Microsoft Google Facebook Twitter Uber Tumblr

# 卷首语

## 薛命灯

秋天是收获的季节，对于极客邦科技来说更是硕果累累的丰收季。九月中旬，2017 CNUTCon 全球运维技术大会在上海光大会展中心成功落下帷幕，并为未能到场但对运维技术怀有渴求之心的同学奉上了两场免费的线上直播。在即将到来的金秋十月，技术大咖云集的 2017 QCon 大会将在上海宝华万豪酒店拉开帷幕。届时，百余位国内外技术专家将济济一堂，为到场的与会者献上数场技术的饕餮盛宴。在 QCon 开幕之前，由 InfoQ 推出的第二本精品图书《硅谷革命：成就苹果公司的疯狂往事》将从 9 月 20 号开始陆续发送到读者手中。

之所以提到《硅谷革命》这本书，是因为它与《中国顶尖技术团队访谈录》有着千丝万缕的“联系”。这本书讲述了三十多年前由乔布斯带领的一个不起眼的团队开发出第一代 Mac 电脑的传奇故事。因为初代 Mac 原型机只是一个孵化器项目，所以显得有点“不起眼”。乔布斯找来了各路英豪，他们不仅是技术天才，还是音乐家、诗人、艺术家，甚至还有动物学家和历史学家。他们热情洋溢，不求回报，只为实现改变世界的梦想，终于在乔布斯的带领下开发出了划时代的

电脑产品。

同样是技术，同样是团队，尽管时代变了，但技术和团队的本质没有变。技术的变革推动了世界的发展，而技术背后需要团队的支撑。三十多年过去了，Mac 从初代原型机演变成了让人爱不释手的 Mac 产品线。技术已经不可同日而语，但技术团队仍然需要极致的热情，始终怀揣改变世界的初心，只有这样才能趟过迷茫的泥潭，不断成长，用技术改变世界。

这一季的技术团队访谈录汇集了知乎、百度、腾讯、360、网易等公司的大咖，他们谈技术，谈团队，谈如何打造高质效的团队，谈如何成为合格的团队带头人。他们谈论的也都是目前十分热门和前沿的技术——人工智能、AIOps、云平台、高性能计算等。

三十多年前的那场革命似乎已经烟消云散，但在技术变革日新月异的今天，它似乎又如影随形。我们无时不刻地在面临技术上的挑战，面临团队合作方面的困惑。如何突破技术瓶颈，走出团队管理的阴霾，又如何在摸索中成长，为改变世界献出一己之力？相信读者一定能够从这一季的访谈录中读到有用的信息。

# Geekbang.

极客邦科技

InfoQ

EGO EXTRA GEEKS' ORGANIZATION  
NETWORKS

StuQ  
斯达克学院

## INTRODUCTION | 极客邦科技简介 »

极客邦科技 是一家 IT 技术与学习服务综合提供商，旗下运营 EGO 社交网络、InfoQ 技术媒体、StuQ 斯达克学院职业教育三大业务品牌。致力于整合全球优质学习资源，帮助技术人和企业成长。

### 企业使命

整合全球优质学习资源，帮助技术人和企业成长

### 企业愿景

打造全球领先的技术人学习和交流的平台

### 企业价值观

公开透明、诚实正直、每日精进  
乐于服务、负责守诺、创新敢为



极客邦科技十年：

- ◎ 2007 年 3 月 极客邦科技创始人兼 CEO 霍泰稳将 InfoQ 引入中国。
- ◎ 2017 年 3 月 极客邦科技十周年，已拥有 EGO（社交网络）、InfoQ（技术媒体）、StuQ 斯达克学院（职业教育）3 条业务线。累计服务国内超过 100 万的技术人，与超过 300 家企业合作，主办超过 50 场技术大会，走进日美，业务覆盖两岸三地。
- 每天，超过 40 万技术人通过 InfoQ 中国微信公众号了解最新技术趋势与最佳技术实践。
- 每天，超过 100 万技术人与极客邦科技旗下垂直矩阵化的新媒体平台交流互动。
- 每月，超过 1000 人在斯达克学院上学习新课程，掌握实用性 IT 新技能。
- 每年，超过 1 万名中高端 IT 人通过 QCon、ArchSummit 等国际性大会学习最新技术实践，了解最前沿技术趋势，结识业内技术大咖。
- 每年，超过千万的独立账号访问 InfoQ 中国的网站了解国际前沿技术与资讯。
- 每年，EGO 有超过 30 场的学习活动，超过 300 位的国内 CTO、架构师、技术 VP 等高端技术管理者通过 EGO 组织建立紧密联系，分享经验，共同成长。

# 目录

**07 专访ImageNet冠军颜水成团队，如何将比赛成果在企业中落地？**

**14 FreeWheel容力：如何打造更高质效的技术团队**

**21 知乎张瑞：机器学习提供个性化定制，未来或推出智能问答产品**

**26 回到网易8个月来的测试团队转型实践**

**36 如何成为一个合格的技术Leader？**

**44 赵成：回顾运维的发展历史，我相信AIOps是必然趋势**

**50 腾讯游戏容器云平台的技术演进之路**

中国顶尖技术团队访谈录 第十季

本期主编 蔡芳芳

流程编辑 丁晓昀

发行人 霍泰稳



**56 深度学习打造精准推荐系统，细说国美互联网AI发展的进击之路**

**62 专访刘文志：并行计算的未来，是让人工智能无处不在**

**74 QQ空间亿万图片存储架构的发展四阶段**

**78 从实践者的角度看 Rust**

**80 落地机器学习前，我们应该思考清楚的几个问题**

**84 如何理解Serverless?**

**87 道器相融，论一个优秀机器学习平台的自我修养**

## 联系我们

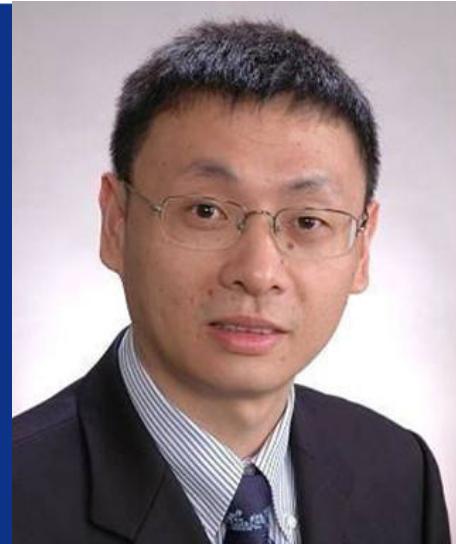
提供反馈 [editors@cn.infoq.com](mailto:editors@cn.infoq.com)

商务合作 [hezuo@geekbang.org](mailto:hezuo@geekbang.org)

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

# 专访ImageNet冠军颜水成团队，如何将比赛成果在企业中落地？

作者 吴少杰, Tina, 陈思



2017 年 7 月 26 日，计算机视觉顶会 CVPR 2017 同期举行的“超越 ILSVRC”Workshop 上，宣布计算机视觉乃至整个人工智能发展史上的里程碑——ImageNet 大规模视觉识别挑战赛于 2017 年正式结束，也就是说 2017 年是 ImageNet 的最后一届。在 2017 年 ImageNet Large Scale Visual Recognition Challenge 2017 (ILSVRC2017) 的收官比赛中，360 公司与新加坡国立大学合作团队拿下了物体定位任务的冠军。InfoQ 因此联系到颜水成团队，进行了这次采访。

ImageNet 竞赛主要分为物体定位（识别）、物体检测、视频物体检测三个大类。在

ImageNet 举行的八年中，物体识别的精度从最初的 71.8% 上升到 97.7%，识别错误率已经远远低于人类的 5.1%。“ImageNet 重新定义了思维模式，虽然很多人关注模型，但 ImageNet 使我们更关注数据，”ImageNet 创始人之一李飞飞说：“数据改变了我们思考的模式。”截至目前，ImageNet 数据集现在超过 1300 万张图片。

## 将ImageNet的成功转化到企业实践

ImageNet 从学界发起，但是赛事一经公布，便有多家科技企业参与，包括：谷歌、百度、微软、

360 公司、商汤、海康威视等等。对于一个特别是针对计算机视觉的比赛而言，其结果对于学术界的影响力是比较明显的。学术界追求的是算法的极限。就是说研究的目的是能不能通过新的算法，以及一些新的 trick，使性能达到能期望得到的最高的上限，我们叫做追求精度极限。这个比赛，每一年都希望在原来的极限基础上再次提升。对于学术界来说，它的特点非常像一个百米赛跑，每次长度是一样的。比如像 ImageNet 竞赛，它的数据从 2010 年到现在，训练集和测试集在分类上基本没有变过，在定位方面也没有变过，那么每年的比赛，都是在做同样的事情，每年都在追求一个新的精度极限，每年极限都有一个更新。

对于工业界来说，当性能在往上提升的时候，就会思考这个精度是不是可以转化到工业界的产品或者服务里面。因为对于有一些问题，如果精度没有达到一个 Threshold（阈值），它就可能不会在工业界里去使用。但如果达到了这个域值，就可以开始去思考，怎么通过算法的精简，让它在工业界具体的产品和服务里面直接用起来。

在 360 内部，我们把人工智能设成四个纬度，一是图像的理解，二是语音的理解，三是语义的理解，四是大数据的理解，或者大数据分析。一个很有趣的事情是，每一次计算机视觉新的算法出现之后，在计算机视觉界的成功会引发这些模型在其他领域进行使用。比如说深度学习里的卷积神经网络，原来是在图像领域用的非常不错，后来有人就把它用到语音和语义里面。计算机视觉错误率的降低，决定它的根基的网络结构，会慢慢的从计算机视觉领域往其他的领域扩展，这样会对整个人工智能领域产生一个整体推动的作用。最早的时候，深度学习的使用点并不是在计

算机视觉领域，而是在语音识别领域。语音识别领域因为问题相对要少，而计算机视觉领域要处理的问题的类型、数据类型差异化非常大，是个非常庞大的领域，因此计算机视觉在整个人工智能领域的声音比其他领域更强大一些。

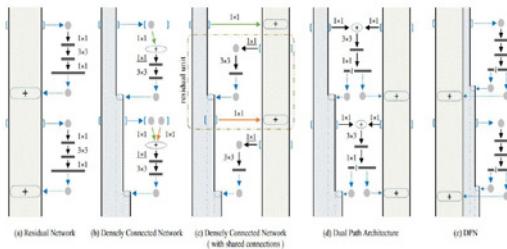
## 逐年叠加的突破和 DPN 模型的提出

自 2010 年以来，ImageNet 竞赛主要包含三类任务，物体定位 (object localization)/识别 (classification)、物体检测 (object detection)、视频中物体检测 (object detection from video)，还有就是已经取消的物体分割任务。如果只是去看所使用的模型，那么慢慢的大家都非常的相像，都有一个基础模型。这个基础模型一般是从物体分类领域出来的，在这个基础之上，还可以把它拓展成物体的检测和物体的分割。那么慢慢的在检测和分割上，除了基础模型外后面要叠加的部分，大家都越来越相似，越来越固化，逐渐变成经典。

大家现在的贡献主要是在这个基础模型上的进展。2015 年微软拿到比赛冠军的时候，他们也是在基础模型上有很大的一个突破。今年的状况也是非常相象的，主要是在基础模型的一个变化，也就是 DPN 模型的提出。至于怎么做检测，大家用的还是比较标准的框架，可能每年会增加一些特别小的 trick。但是这些 trick 并不是根本性的，根本性的还在基础模型的变化上。

在这次 ImageNet 竞赛中，颜水成团队提出了双通道网络 (Dual Path Network) 模型。在 ImageNet-1k 分类任务中，DPN 网络不仅提高了准确率，还将 200 层 ResNet 的计算量降低了 57%，将最好的 ResNeXt (64x4d) 的计算量降低

了 25%。



这两年有两个非常重要的网络结构，一个是在 2016 年的最佳论文，ResNet；一个是 2017 年获得 CVPR 的 Best paper 的 DenseNet。大家在早期研究 DenseNet 的时候，DenseNet 还并没有拿 Best paper。我们的核心成员在研究这两个网络之间的相关性，发现 ResNet 可以解释成 DenseNet 的一个特例，2016 年的 Best paper 是今年的 2017 年的 Best paper 的一个特例。

它的分析是基于我们以前的 Network-in-Network (NiN) 里的  $1 \times 1$  的卷积。差别是里面有一个  $1 \times 1$  的卷积，如果在不同的层，用的是同样的参数，DenseNet 就能转化成 ResNet。在这个分析的基础上，我们认为 ResNet 和 DenseNet，虽然是个特例，但是初衷各有优劣。ResNet 不停的去 refine 改善和修正它的特征，但是 DenseNet 相当于不断的去探索新的特征。两种策略都非常重要。举个例子，就是说好像一个公司要往前去发展，一方面你的老员工要不停的自我修炼，要自我提升，这个有点像 ResNet；另一方面，又要不停的去吸收新的血液进来，这些新的成员有新的特性，那么这两部分合在一起，最终的话，能让公司可以稳健的往前走。

基于这种思想，我们就提出一种网络结构，一方面的话，要对原有的一部分特征进行自我提升，叫做 refinement；另外方面，让一部分网络探索新的特征，然后这两部分特征融合在一起，一层一层的往前走。这地方有一个有趣的问题，有人在问我们，ResNet 既然是 DenseNet 一种特例，那么现在是把这两种网络的特性融合在一起了，那么新的网络是不是一个新的 DenseNet？这个问题很有意思。我和我的学生陈云鹏，想了一个晚上，最后总结，得到的新的网络不能解释成 DenseNet。虽然是把两个网络融合在一起得到一个新的网络，这个网络融合在一起，为什么不是新的 DenseNet？我们举个例子，你有一个等边直角三角形，如 DenseNet 就是一个任意方向的等边直角三角形，ResNet 是某一个特定方向的一个等边直角三角形，两个等边直角三角形拼接好会还是个等边直角三角形，还是 DenseNet。但是我们虽然 ResNet 可以解释成是 DenseNet 一个特例，但是它们两个融合在一起的时候是有一个错位的。类比，两个等边直角三角形如果用长边去做对接，那么出来就是一个正方形，就不再是等边的直角三角形了。DPN 用到了 ResNet 和 DenseNet 的思想，结构拼接的方式比较特殊，所以它出来的网络结构化就已经无法解释成 DenseNet 了。它变成了一种新的网络结构。优势就是充分利用到了修正和改进的特征以及新特征的探索两方面。

## ImageNet 的价值和模型关键节点

不只是 ImageNet，还包括做人脸检测和人脸定位，都有各种各样的 benchmark。有一些把测试的样本的标注已经告诉你了，那么就可以不

停的不受次数约束地去测试各种算法；还有一种，像 ImageNet 还有一些人脸识别比赛的话，它提供训练数据，校验数据，但是测试数据只提供原始数据，不提供标签。你需要把结果发送到一个服务器上，这个服务器会做结果评价，同时会限制一段时间内能提交的次数，防止不停的去刷参数。如果能不受限制地刷参数，就变得纯粹是一种体力劳动，不是真正意义上的智力活动了。

很多团队，不论是学术界，还是工业界，都希望在这些比赛和 benchmark 上去拿到很好的结果和名次。基于不同的诉求，一种是对他们，比如说融资，有很大的帮助，另一种是为了就是秀肌肉，展现自己的强大实力。我觉得这些都是没有错的。但是刷榜本身是没有价值的，真正的价值在于，刷完榜之后，有没有为这个 community，为所在的领域留下有 Insight 的东西。在 ImageNet 领域，在不同的时间点，留下了很多很不错的基础模型。这些基础模型对于整个领域都是非常有价值的。

最早的一个时间节点是 2012 年，AlexNet 第一次把传统的模型、传统的算法转换到深度学习。2013 年我们参加了 ImageNet 比赛，我们在分类里面取得了第二名。我们当年提出了一个新的模型结构，叫做 Network in Network，实际上我们那年是没有来得及用，但是我们那年在参加 Workshop 的时候，就把 Network in Network 介绍了。2014 年有两个非常重要的模型，一个是 GoogleNet，一个是 VGG Net。GoogleNet 作者把他们的模型叫做 Network in Network in Network，受到 Network in Network 一些启发，特别是大量利用了 Network in Network 中提出的 1x1 卷积。2015 年是微软的 ResNet。2017 年 DPN 的总体效果也挺不

错，不少学者在自己领域实验的结果来看具备不错的普适性。

## 深度学习的低门槛已经能面向普通开发者

深度学习发展到现在，特别在计算机视觉里，大部分东西都已经非常成熟了。甚至可以说现在一个本科一年级学生，他就可以训练出一个一个车辆检测器，或者人脸检测器。也就是说深度学习的门槛不是非常高了。如果说一个企业有一个具体的业务场景，希望得到还不差的性能的话，那么已经没有必要说要拿一个 PhD 去做这件事情了。可以利用开源的平台和一台 GPU 机器，甚至可以运用云计算，把数据放进去，就可以训练出比较不错的模型出来。

那么深度学习的研究，我觉得有两个纬度，一种是追求精度的极限，一种是追求用户体验的极限。追求精度的极限的话，要做的事情是去思考，当前最好的模型在解决具体问题上还有哪些 weakness（弱点）。在这个基础上，再去提出新的模型的结构，在基础模型上去做提升。但是这个非常的困难，为什么？比如说，要想再去超越 ImageNet 比赛最好的性能，所需要的计算机的资源是非常的巨大的，这是第一点。第二点，我觉得对于无论是 ImageNet，还是说人脸识别比赛，通过纯粹算法能提升的精度已经非常少了，基本上都是个位数，或者零点几的提升。那么从这个角度来说，我觉得我们需要有更难的，更有挑战性的数据型集。这也是今年 ImageNet 比赛到了它的收官之战的一个主要原因。

在这个 ImageNet 之前，还有一个比赛，叫做 PASCAL VOC，是欧盟组织的一个比赛，进行了 8 年左右。我们参加了 PASCAL VOC 后三届

的比赛。PASCAL VOC 比赛在 2012 年也进入收官之战，我们那年参加了比赛，当时拿了两项比赛的第一。PASCAL VOC 比赛当时进入收官之战的一个主要原因，是因为它的图像标签的数目只有 20 个，跟真实的应用场景有一定的差距。ImageNet 竞赛，现在也要进入收官之战一个主要原因，是因为它的分类的数据集每张图像上只有一个主要物体，跟真实的图片的状况有很大的差距。一般拍的照片不只是一个主要的物体，应该是有很多的 label，是多标签的图像理解。那么 ImageNet 之所以没有弄成非常大的，多标签的图像分类的一个主要原因，是因为标注量会有大幅的提升。

## GAN 和增强学习是深度学习的两个热点

GAN 和增强学习，是当前深度学习的两个热点。增强学习在 Game 领域非常成功，因为它需要有一个 Reward 机制，这个机制在 Game 里面是直接的，从最终的结果里直接能判断出来，但是视觉里面 Reward 的获取不是特别的自然。但是 GAN 的话，对于计算机视觉领域的发展是有基础性的推动作用，特别是对于这种 pixels to pixels 问题，从输入的图像大小，到输出的结果图片的大小是一样的时候。像分割、风格化、超分辨率这些问题，GAN 是非常好的。另外一方面，最近 GAN 用的非常成功的一个地方是用 GAN 去生成一些虚拟图像，这是非常有价值的。我觉得 GAN 还能火几年，还有很多的问题可以做。

## 计算机视觉技术在 360 的落地

360 最核心的业务是安全。安全一直是整个公司的灵魂和基因，包括 PC 的安全，移动设备

上的安全，以及企业的安全。安全产品带来巨大的流量，这个流量就可以直接推动内容和服务的产品，比如 360 导航，360 搜索，360 手机助手，信息流、花椒直播，AR 相机，短视频等等。另外一块，老周从 2012 年开始专注智能硬件，物联网方向。360 专注在三个方面，一个方面是穿戴设备，主要是面对儿童的。如 360 儿童手表，已经第六代了，销量应该是达到了 500 多万台。第二个，家居安全，主要是包括 360 智能摄像头等。第三个是面向出行的安全，主要两款产品，一款产品是 360 的行车记录仪，还有 360 的后视镜。

360 的计算机视觉应用场景，主要是在内容产品和物联网产品。内容上除了支持 360 的图像搜索；还有 360 的短视频，以及花椒直播和相机。用计算机视觉去对图像、视频进行分析、理解，分析出来的内容会作为推荐和搜索的依据。直播和相机则主要是用计算机视觉技术，做美颜、人脸的分析，让拍摄的过程更加好玩，更加有趣。在智能硬件这块的计算机视觉技术，一方面是人脸的技术，另一方面是人体的技术。人脸的技术主要是在监控摄像头和儿童机器人，使它能够识别家人，识别用户。在出行安全领域，后视镜里面有辅助驾驶的功能，对车辆和车道线进行检测，实现前车碰撞报警，车道偏移报警，前车起动报警。

## 将人工智能跟信息安全相结合

360 很早就把人工智能和信息安全相结合，行业内第一个具有人工智能技术的杀毒引擎就是 360 的 QVM 引擎，QVM 引擎采用人工智能算法——支持向量机，具备“自学习、自进化”能力，无需频繁升级特征库就能免疫 90% 以上的加壳

和变种病毒，从根本上攻克了前两代杀毒引擎“不升级病毒库就杀不了新病毒”的技术难题。现在 360 的安全部门也有一个团队，专门利用深度学习解决很多安全相关的问题。比如说在 2015 年，他们在 Blackhat 大会上曾经有好几篇文章发表了，用深度学习 Data driven 方式去做安全相关的问题。人工智能跟信息安全相结合，肯定是一个趋势。今年 360 会继续去举办互联网安全大会，里面会有一个专门的 keynote speech 讲人工智能和信息安全结合的问题。

## 计算机视觉未来发展和面临的挑战

对于计算机视觉发展的挑战和前景，团队里的董健回复说，第一个方面，现在很多成功的计算机视觉算法主要是在监督学习领域，需要进行大量的数据标注，但是计算机视觉如果想得到更广泛的应用，还是要考虑更好的利用非监督数据。第二点，现在计算机视觉的算法，在计算资源不受限的环境下，比如有 GPU 集群的情况下，性能是比较有保证的。但我们在很多实际应用中，比如在嵌入式设备中，或者云端部署但用户量非常巨大的情况下，对性能和速度都有非常高的要求。如何将算法进行压缩和加速，也是一个比较大的挑战。当然，这个可能会从软件角度和硬件的角度同时进行推进。

而颜水成对小样本问题给了我们进行了解释。计算机视觉里面，会有些小样本问题，小样本的问题怎么去解决呢？小样本会有两种情况，一种是样本少，还有一种情况是没有标注数据，但无标注的样本很多。对于第二种，无监督的数据怎么去使用就会非常有价值，就像董健刚才说的，这块会是研究和应用的一个重点。怎么用有

两种，一种是在训练模型之前，可以用这种无监督数据做 pre-training，还有一种跟人是非常像的，人对世界当有了一个初步的认识之后，在跟物理世界接触的过程中，在识别时脑海里已有这种识别的模型，在这个模型部署之后，还需要有一种自我学习和自我更新的能力。这种数据没有提供标注，可以认为是一种无监督的方式。

另外，计算机视觉的落地，跟学术界不一样。学术界可以在资源完全不受限的情况下，去做这种事情，去达到精度的极限。但是在工业界，计算机视觉算法的落地，无论是放在云端，还是本地，如嵌入式设备或者手机的设备，降低计算的复杂度，都是有价值的。比如说，假设如果能将人脸识别的计算复杂度降低一半，只要部署一半的资源就可以完成任务，对于公司来说，收益就会有大幅度的提升。除了去追求算法的精度，降低模型计算的复杂度肯定是要重点思考的方向。

## 深度学习实践建议

**实做深度学习图像处理任务时，在实际的应用场景中，Caffe/Caffe2、Tensorflow、PyTorch/Torch 这些工具如何选择，有什么好的建议？**

**董健：**这个问题要看具体的应用场景。如果是学术界，或者是偏研究的应用，优先推荐选择 Tensorflow，或者 PyTorch，因为语言是大家比较常用的 Python，开发会比较简单。而且整个网络架构的设计非常灵活，如果想设计一个新的网络结构，代码的开发量会比较小，开发速度会非常快。如果要在实际工业界使用，要看具体的场景，如果是在云端的一个服务，或者说服务的压力不是非常大的话，那像 Tensorflow/mxnet 也是比较合适的，因为它整体部署起来比

较容易，效率各方面也能满足要求。如果在嵌入式平台上开发，因为 Tensorflow 和 PyTorch 相对复杂，速度会更慢一点，而且由于代码量大，导致库的大小也比较大。因此在移动端，像 Caffe/Caffe2 这样比较简单的库更加合适。

### 对现在各个工具平台的 model zoo，是用 pre-model（预训练模型），还是自我训练模型，有什么建议？

**董健：**这个问题现在基本上已经达成共识了，就是说如果在这个问题上，你的训练数据已经非常多的话，那你可以直接训练。但是如果你训练数据没有那么多，大家的一个通用做法是，先在一些标准库上，比如 ImageNet，进行预训练，再用预训练模型在具体的问题上进行 fine-tuning。这是一个比较流行也有效的措施。另外，实际使用当中的话，如果你要解决一个实际问题，

通常是要根据具体应用设计专门的模型。这个时候，你可能也没有办法直接用网上预训练的模型，但是一般也会进行预训练的操作。

**实际生产环境，训练样本偏少，要提高识别准确率，应该在哪些方面多做工作，人工收集样本数据还是通过算法模型标注样本数据？有什么好的建议。**

**董健：**这个也是现在学术界努力解决的一个问题，我们实际在工业使用当中，一个纯 unsupervised 的问题还是比较难解决的，我们更多的是通过算法加速数据标注。具体来说，有点类似于 Active Learning 的思想，先用算法对数据进行一个标注，之后再用人工进行筛查反馈，对数据进行一个修正，目前可能说想完全通过非监督学习来提高精度，难度还是比较大的。

**颜水成**，360 公司技术副总裁，首席科学家，人工智能研究院院长。在加入 360 之前，在新加坡国立大学做计算机视觉和机器学习的研究，在 2015 年年底加入 360。新加坡国立大学终身教职，IEEE Fellow，IAPR Fellow 及 ACM 杰出科学家。他的主要研究领域是计算机视觉、机器学习与多媒体分析，发表近 500 篇高质量学术论文，论文引用过 3 万次，H-index 74。2014、2015、2016 三次入选全球高引用学者。

**董健**，360 高级数据科学家，前 Amazon 研究科学家。目前主要关注深度学习、强化学习、计算机视觉等方面的科学和技术创新，拥有丰富的大数据、计算机视觉经验。曾经多次领队参加 Pascal VOC、ImageNet 等世界著名人工智能竞赛并获得冠军。博士期间在顶级国际学术会议和杂志上发表过多篇学术论文。从 2015 年年底加入 360 至今，董健作为主要技术人员参与并领导了多个计算机视觉和大数据项目。

# FreeWheel容力：如何打造 更高质效的技术团队

作者 曹倩芸



## 编者按

技术团队的管理包罗万象。但归结到底，无非在于二字：道、术。

听过很多人讲二者间的关联，却大多未全然说透。道与术的关系本质上就是一个指导思想与具体手段的关系，是心法与招式的关系。从技术角度来看，管理中的“道理”、“心法”或可界定为技术定位、技术与业务的融合、企业文化、技术与哲学等；“方法”和“招式”，则就如技术团队组织架构、产品开发流程、制度规范的建立、架构设计等。

怎样将“心法”与“招式”融合在一套特殊的理念和决策之下，或是困惑多数技术管理者的一道难关。带着对如何打造顶尖技术团队、如何

在技术、文化以及技术人成长间搭建更优桥梁、如何培养团队荣誉感及价值感等问题，InfoQ 专访了 FreeWheel 高级副总裁容力及其所领导的技术团队管理者（技术副总裁党政法、技术副总裁王强），了解在这样一支跨国团队协同合作的环境背景下，作为 300+ 人团队的总负责人，他是如何塑造高效敏捷的行动力、团队方向感、追逐感、凝聚力，以使庞大（甚至是跨界、跨时区）的队伍能拥有更好的创新力和生命力。

曾在知乎上看过一个提问：“互联网公司，如何管理一个 8-10 人的技术团队？”，下列的跟评有着林林总总各式不同的看法，其中一大致列了三条最简单也最复杂的见解，即了解人、了解事和了解业务。所以，我们和 FreeWheel 技术团队的交谈也从这几点开始。



## 万丈高楼平地起 “合理扩张” 与 “Hire the Best”

公司高速发展的时候非常需要适时的管理。从刚刚成立扩展到几百人的规模，这是每个公司发展需要经历的一个很大的槛。有些企业疯狂式的增长，是对现实性的考虑模糊。我（FreeWheel 高级副总裁容力，以下均以第一人称“我”代之）观察到过去太多大公司人员的迅猛增长，并非是由公司的效益或业务需求所驱动的，而是取决于领导层的话语权。

FreeWheel 在 2007 年刚刚成立时只有 10 个人左右，2015 年 3 月我加入时已上升至 150 余人，而截至目前，FreeWheel 又扩增了一倍，达到 300 余人规模。经历十余年的发展，整个创业和探索的过程非常曲折和艰辛。但一直以来，团队的扩建都是由业务所驱动的（FreeWheel 每年

保持大约 70% 的营收增长率），因此这决定着更为良性的增长动因和态势。

人员持续扩张的背后，“Hire the Best”是在招贤纳士时秉持的最重要的基准。何谓 Best，如何定义？表层意义上来说，Best 就是顶尖的，具体分析主要体现在两个方面：一个是在校招中，学历与毕业院校的卓越性，另一个更重要的是在技术上的扎实积淀和积累，放在不同的招聘环境下具有不同的侧重点。但同时，Hire the Best 的难点在于，你不能只看他现在会什么，而是要预测他将来是否能更快地学到更多。技术人员的学习能力，其个人长期的潜质和潜力应该在面试环节得到更多的加分。

我个人有一个理念，不管是在应聘还是在招聘的时候，都会秉承一个非常浅显的道理：我们看待一个人适合不适合这家公司就看三点，第一，

是这个人能为公司带来什么价值；第二，公司能带给他什么，双方是互利的过程，仅仅看这个人给公司带来什么，不看公司带给他什么，这个交易也谈不成；第三，这个人能够在公司做多久。我们是对人才的投资，工作也是他生命中贡献时间和青春很重要的一部分，所以是否能够长期在一家企业效力很重要。因行业而异，因公司而定。

## 拥有良好工程师基因团队的三点练就要素

### 管理人员必须从技术一线提拔而来

如何从一线的工程师转成技术管理者，对个人和工程师团队文化来说都是非常重要的一个环节。我非常看重的一点是，技术公司的管理人员一定需要从技术第一线提拔而来，这样才能让公司保持工程师团队文化，而且这种文化才具有与生俱来的某种技术特性。

我当时来 FreeWheel 就跟一线管理人员说，如果你的技术能力不能让你在团队里服众的话，那你还能给团队带来什么。在我的定义和 FreeWheel 的文化中，转变为团队领导的人必须要在他的团队里具有最顶尖的技术水平。我们一般会以 10 人左右的团队为一个评价标准，如果他是技术大拿，只要他说了别人就觉得有信心，愿意按照他说的去做，那么他才能成为一个合格的团队领导者。很多硅谷的互联网公司也是这样，并不是按级别的高低来选择最适合的管理者。

### 培育一种开放的文化：信息和思维双维度

我曾在微软和雅虎工作了将近十年，从最早的编程开始做起，到做一线的技术管理，再到管理 100 人、300 人的团队，这种锻炼是一步一步做起来的。十多年里，从一线的管理逐渐转变到

管理更大的团队，我也有了更扎实的体会。关于如何建设团队；如何做长期的规划；对于一个团队，以至于一个比较大的团队组织，怎样获得长期且稳定发展；如何在良性竞争中胜出等问题，微软和雅虎都为我提供了很好的学习平台，所以在加入 FreeWheel 之后，可以很自然地将这些理念和思路应用在具体的技术管理场景中。

在雅虎期间，我能感觉到那种硅谷的沸腾的氛围，而且在硅谷不同公司之间的交流也非常频繁。在硅谷，不管是大公司还是小公司，大家就像是在一起创业的大家庭，时间长了就形成了一种工程师文化，说的简单点是自由、平等、开放。这种工程师文化会让每个人都持续保持进取的态度，很少有人抱怨，心态也会更加开放。

都说技术老大的角色决定了一家公司技术团队建设的模式，FreeWheel 联合创始人兼 CTO Diane 在工程师开放文化的构建上起了非常关键的带头作用，我们的开放主要体现为在信息维度的充分共享和在思维维度的创造激发。整个公司发展中，技术层面和商业模式层面的信息，可以在领导层和员工之间做到充分的沟通和共享，使得全员在任务处理上能更好地把握整体方向、理清事情的优先级，做对公司最重要的事情。另外，大家做事情都是本着平等开放的原则，不会因为说话的人不同，就对他提出的问题或者方案有不同的态度，不因人废言，更不因言废人。这样做的目的就是要激发大家的积极性，充分发挥每一位员工的创造力。

### 对Engineering Excellence的追求

追求 Engineering Excellence，是近期 FreeWheel 整个工程师团队的最大变化。在公司整体度过了生存期的挑战并进入到加速生长期时，我们要关注的事情，不再是到处救火，而是

要追求卓越，要打造一个可以在未来几年里，支撑业务发展的优秀技术平台。在这个新的目标下，FreeWheel 工程师团队也发生了不少变化，包括对 Full Life Cycle Engineer 理念的倡导，对 CI/CD（持续集成 / 持续交付）等高效开发流程的探索和精进等。

## 敏捷开发模式在工程师团队的实践与落地

敏捷开发模式跟传统的开发差别之一是对变化的拥抱。传统的开发相对会拒绝变化，即计划制定好之后不希望有任何变化，有变化会对开发流程造成很大的影响。敏捷文化是拥抱变化，当有问题发生的时候需要会主动做调整，当然这对团队的组织和行为方式也带来了很大挑战。这里，FreeWheel 有一些比较好的实践经验可以分享。

**第一是团队的组织。**FreeWheel 一开始按照敏捷开发的原则将团队组织成 Scrum，在这种管理方式下，所有相关人员（主要是测试人员、开发人员，和在美国的产品团队）会形成一个比较紧密的团队，消除相互间的壁垒。所有人从一开始就会融入到产品的设计、开发、测试里去，通过快速反馈和及时沟通，能最迅速地解决项目过程中各种问题。

**第二是团队的管理。**敏捷开发与传统模式相比，最显著的特点就是人和流程的关系有很大不同。传统开发中会制定一个固定的流程和周密的计划，人被添在了这个流程和计划中。敏捷开发的变化多、迭代快，如果套用一种管理模式并强加给团队，往往会造成生产力的倒退。所以管理的方式应该是给团队更大的自主性，注重引导而非控制。一个公司往往会有多个敏捷团队，我们最终是让各个团队用他们的聪明才智解决问题，

但同时各个团队之间又可以互通有无、取长补短。有的时候你会是第一个踩坑的人，但这个踩坑的过程是有价值的，得到的经验教训可以跟大家分享。

**第三是做事的方式。**当把任务拆分的更细时，反馈就能更及时。如果有一大块的东西挡在你的工作流里，会对你的敏捷性造成障碍。所以我们一开始就会对所做的事情进行拆分，把任务拆分得更细，使得任务更容易被排期。同时在技术上实现 CI / CD，对产品持续不断地做测试，并且把它集成到研发流程中。当工程师做了某些改动后，整个系统会立刻产生结果评价，告知你做的是对还是错，然后大家基于这个结果再看如何继续推进。

## Scaled Agile Framework 模式的尝试

与此同时，上面提到的“Hire the Best”理念也会对团队建设和管理带来一定挑战。如果团队人员的水平和能力呈现为梯队型，自然而然管理会相对容易；但如果大家都处于较为平均的基线范围内，就会面临更多协调、平衡及取舍方面的工作。最近 FreeWheel 正在尝试 SAFE，即将团队都分成不同的 squad，一个 squad 即为对产品有贡献的垂直的功能性团队，需要挑选并重组在前端、后端、数据库方面具有差异化优势的小组成员。但将优秀的成员都放在一起做一件事很容易有人做的多，有人做的少，有人满意，有人不满意，这就跟项目管理、软技能训练等相关。

通过实行敏捷模式，工程师团队的整体趋势会越来越扁平化，但是扁平化主要还是发生在在一个模块内部。比如，目前 FreeWheel 中国的工程师主要被划分为五个分支技术团队——

AdServing、Forecasting、Reporting、UI 和 OPS，其中，AdServing 模块内部被大概分成了 6、7 个小型团队；Forecasting 被分成 3 个小型团队。其他四大模块情况也类似。在团队组织上，部分小型团队成员可以直接汇报给一线研发经理，例如，UI（负责核心业务系统的研发与测试）模块的 80 人团队（北京分部）大概有 6 个研发经理，他们各自领导着十人左右的团队。因此，分支技术团队总负责人和普通工程师之间只有一级，向上的沟通和汇报会更加通畅。而且，负责管理的同事有更多的机会直接与小型业务开发团队进行协作，第一手掌握他们真实的想法、他们的困难和反馈意见，以便缩短做决策的时间周期。

## 对Full Life Cycle Engineer理念的倡导

对于 Full Life Cycle Engineer，现在业界有两种声音，一种是 Full Stack（全栈）Engineer，指能够掌握前端、能写 Web，能写后端服务器，还能开发 mobile 程序等，要掌握不同的技术。这种声音存在一定的争议。另外一种方向是 Full Life Cycle Engineer。比如写 C++ 的就不需要掌握 web，但要能对使用 C++ 开发软件的整个生命周期熟稔于心：当新的功能要求提交出来时能用 C++ 对它进行设计，写完之后能够用相关的工具测试并将服务进行发布，后续还需要支持服务的运维。举个很简单的例子，测试、运维的环节常常被技术管理者忽略，随之而来的问题便是产品质量不可控、BUG 一堆、发布成功率低、运维人为事故频发。但对 Full Life Cycle Engineer 理念的倡导可以有效减少此类问题的发生。

因此，FreeWheel 非常倡导 Full Life

Cycle Engineer，其直接优势在于可以消除上下游间明显的界限。比如一开始设计的时候就会将测试的实施方案、后期发布、运维上的问题一并考虑进去，而不是执行一半之后再去想怎么“填坑”。这是敏捷模式下最为可行的方式，如果整个软件交付过程中还要上下游切换，沟通成本就会很高，整个团队不可能敏捷起来。

## 最快速地定位技术问题出现的根源

研发团队之间的协作非常重要，协作和沟通的效率很多时候都是解决复杂技术问题时的必备条件。开发过程中，如果我们在上线环节亦或开发测试环境中发现了一个具体的技术问题，不同的研发团队负责人或该领域的资深专家会组成一个临时小团队，不断地调研，分析和研究后认定问题根源并划定应该由哪个团队、哪个工程师具体跟进和解决。解决方案落地且开始执行后，大家还会回过头来进一步验证，确定目前的解决方案是否有其他风险存在。

## 轮岗制的跨地域协作

对 FreeWheel 而言，最大的挑战之一是跨地域、跨时区的沟通与协作。FreeWheel 除了在美国、中国、法国有研发机构以外，在欧、美、亚洲的多个国家也会有自己的办公室。这些国家的团队规模及成熟度较高，但都存在较大的时差。

相比通过电话、电子邮件进行跨时区的交流，我们会尽可能地采用很多其他的方式优化跨时区、跨地域的合作和沟通。比如北京的各模块的研发团队会持续派轮岗人员，去美国等地不同的实验室和不同部门，与产品经理在一起紧密地合作一到三个月。这样做的好处之一就是当产品经

理面对更多的客户、分析客户需求的时候能得到研发团队第一手的支持，比如从技术角度看业务设计是否合理、能否实现、实现难度大小等。

另外，美国的产品经理会每个季度飞到北京来与这里的同事工作 1-2 周的时间商议未来 3-4 个月的研发及项目规划。这类协同工作的目的都是尽可能地减少跨时区、跨地域沟通的难度和障碍。如果团队之间没有很好的沟通，做完以后发现有很多问题，但由于一开始没有识别出来，最后就会演化得愈发严重。

## 让每个人感觉到自己的价值所在

人才培养更关注个体，团队建设更关注集体。团队一方面要做事，另一方面要育人，人才是团队的核心资产。在团队中可能只是很微小的一粒存在，你也必须要给所有技术人以存在感和价值感，让他们能看到自己的付出能改变产品、改变公司，甚至改变整个行业的走向。

## 保持团队的方向感，让团队成员知道自己在做什么，将来又要做什么，能感觉到自己的价值所在

在 FreeWheel，不论每个同事的职务或工作职责如何，技术管理层都会首先注重帮助所有人在公司找准自己的定位，也会帮助他们分析清楚自己的长处和短板，并有针对性地制定相应的工作计划及提高个人能力和适应职业发展的规划，帮助所有人找到他对公司的价值所在。这样做一方面会让每一位同事在工作上形成满足感，另外一方面也保证了他做出的贡献和体现的价值能被公司所认可，因而能有更好的职业发展机遇。

## 保持团队的进步感，让团队成员感觉到自己每隔一段时间都能学到新的东西，从而值得

## 为之付出努力

对工程师来说，发挥价值的地方仍在于与产品的强联系。我们非常提倡让技术人深入到第一线工作中，让大家去直面解答客户的问题。在 FreeWheel，销售和客户服务团队是直接做售前、售后服务的人，每个季度聚集到北京参加 PI Planning (Program Increment Planning)，制定下一季的产品计划，从而去打破技术人员、产品经理、销售这一套人员固定模式上的隔阂，让工程师团队能感觉到自己做的东西所发生的变化、对用户所产生的影响，直接参与到用户的意见反馈环节中去。

另外，随着业务的发展和体量的增大，FreeWheel 也正在实行大规模的技术重构工作。当每天的工作中都有可能面临到新的技术挑战时，就能更好地保持团队的进步感。

以 UI 模块团队为例，从 2007 年到目前，前端技术框架经历了多次演变，去年开始 UI 团队（同时也包括其他团队）都在推行微服务化 SOA 架构。后端的业务逻辑会通过 Golang 服务封装成一个一个不同粒度的微服务，这样我们整个前端的框架会更多专注在业务交互上，跟核心业务逻辑直接相关的实现都会封装在底层的微服务框架上。但对于像 Go（包括我们前端目前正在使用的 React.js 框架）都是比较新的技术框架，会在技术细节方面面临更多的挑战，很难通过参考别人的经验就能获得有意义或有帮助的答案。所以我们更多是依赖于自己的研发团队，深入地具体分析某一些技术问题产生的原因：为什么在我们这样的产品环境下会发生；能不能做一些实验，尝试找到一些解决办法；论证我们的解决办法，确保不会带来其他的衍生问题；对我们的系统稳定性、可靠性是否造成影响等。



很多的问题都是挑战，而克服挑战、解决问题的过程才能让大家感受到每天的进步以及明天未知的新奇。

## 保证团队成员的归属感和自豪感，这样的团队才有凝聚力

FreeWheel 常常说我们的企业文化是 Proudly Unique、Deeply Caring，还有 Purposeful，虽然是几个形容词，但是在日常工作中会遇到很多事情，大家会经常在一起谈解决问题的方法，遇到矛盾的时候又怎么解决，这会让所有人感觉与他人连同在一起，是有意义的存在。

另外，企业的定位和愿景同样会对技术人员的价值观形成很大的影响。我入职前，跟

FreeWheel 的 Co-CEO 进行了一轮电话面试，谈公司的发展前景时，他用十年前苹果和诺基亚做了类比的例子，并说：“虽然在公司规模上，我们在业界和主要的竞争对手相比非常弱小，但是如果把竞争对手比作诺基亚（那个时候），那我们就应该是苹果。”举这个例子的目的在于说明，即使目前的你尚且弱小，但只要你的产品有非常清晰的、适合市场发展的战略，也一定能够打败当时占领市场 80% 份额的巨无霸。

在这样的目标之下，我们也相信，优秀的工程师会逐渐认知到企业和自身的使命，认知到他在做的这项技术的前景与价值。

**容力**，在加入 FreeWheel 之前，容力是雅虎北京研发中心的资深工程总监，负责管理该中心所有显示广告产品线的研发工作。在雅虎之前，他服务于微软公司，担任资深开发经理一职，在公司的用户行为研究部门带领由软件开发工程师与研究科学家共同组成的团队，负责搜索广告、基于用户行为的广告定向、基于搜索的广告重定向以及内容广告等的产品开发。容力拥有加拿大滑铁卢大学的博士学位。

# 知乎张瑞：机器学习提供个性化定制，未来或推出智能问答产品

作者 陈思



## 编者按

知乎，中文互联网最大的知识社交平台，用户通过这个平台，彼此之间分享经验、交流知识。从 2010 年发展至今，知乎已经拥有超过 8400 万用户。如此庞大的用户群体，每天都会产生十分巨大的数据，如果只依靠人工管理，将会是一个巨大的挑战。知乎将人工智能引用在各个不同的应用场景，InfoQ 记者来到知乎，采访到知乎机器学习团队负责人张瑞，他将为我们解密机器学习如何为知乎提供高效运营管理。

## 知乎与机器学习

关于人工智能的产品，张瑞说：“大家在讲

人工智能的时候，可能看到的最为显眼的东西，或者是说最能抓人眼球的东西，是直接把机器学习技术包装成一个产品去卖。比如说无人车，它直接做的东西就是无人驾驶；还有比如说面部识别技术，去做监控，或者安防。这些东西你能直接看到，这就是一个人工智能的产品。”但是对于知乎，张瑞认为知乎首先是一个用户体验导向的产品，为了保障用户体验，实际上用到大量的人工智能，或者机器学习的技术是以 AI 为支撑的一个产品。

据了解，知乎很早就开始做机器学习方面的工作，比如说知乎的搜索和推荐。搜索和推荐里面有很多的策略，实际上是用机器学习和数据挖掘的技术。在 2016 年之前，机器学习 case by

case 地应用在一些业务场景上面，去解决业务面临的问题。随着知乎的产品发布的节奏逐渐加快，同时有许多业务场景，需要用到机器学习技术，去提升用户体验，提高社区运营效率。从那年之后，知乎的产品发布的节奏逐渐加快，同时有许多业务场景，需要用到机器学习技术，去解决一些用户体验的问题，或者社区管理的问题。这个时候，再去纯粹依靠人力进行产品决策和社区运营，就会非常的浪费工程师的人力资源。

“所以我们在 16 年的时候，我们就开始组建一个统一的机器学习和数据挖掘的团队，然后大家会在一起，比如说我们构建统一的推荐引擎，然后把这个推荐引擎用到各个业务上面去，比如我们的社区，知识付费产品等，然后会用到各个产品上去，来节省我们工程师的人力，提高我们做事的效率，以及最重要的，不断提升用户体验。”张瑞说。

使用机器学习技术，可以更加精细地对用户和内容进行建模，促进内容生产和内容分发的效率。具体说来，可以把知乎对机器学习的应用划分为 6 大场景，也就是用户画像、内容分析、排序、推荐、商业化和社区管理。

## 机器学习的应用场景

### 推荐

知乎有很多的推荐业务场景，比如邀请回答，从技术层面来讲，其实就是推荐问题。在知识社区里面，这个问题其实被称为问题路由，知乎技术团队做了很多的研究，怎么去把问题路由做得更加高效。张瑞说，知乎在做问题路由的时候，其实有很多机器学习，或者数据挖掘的技术在里面：比如说首先要知道，一个用户是不是适合回答某一个问题，或者说他回答某一个问题的频率

是怎么样的，然后它回答出来，回答答案的质量是怎么样的。对于用户，机器学习会做的这样一个标签，或者说预测分析。

在问题的方面，机器学习会把问题进行基本的理解，比如说这个问题是属于哪个领域的，它的难度是怎么样的，然后去匹配到这个领域下面，适合回答这个问题的人。从回答者的角度来看，既然知道他适合回答哪方面的问题，在等待回答的页面上就会给他推送一些他可能适合回答的问题。

对于问题路由，人和机器，这两方面的表示，现在还是习惯于把它表示成一种，人能工理解的标签。张瑞举例说：“比如说用户 A，他是互联网底的一个权威的回答者，他也比较愿意去分享他在互联网的内容。然后问题 A，恰好是互联网下面一个比较深度的问题，可以交给用户 A 来回答。我们是根据这种，人能工理解的标签，去做的匹配。”而现在，张瑞的团队希望加入一些更深度的东西：假如说一个人，他可以表示成一组向量，或者说现在比较火的技术，嵌入式表示。这个象量人看上去可能只是一串数字，完全不能理解，但是机器可以去理解它，之后做出更好的推荐，“我们也在努力的往这方面去做一些积极的尝试。”

### 用户画像

在张瑞看来，用户画像实际上是一切个性化策略的基础。个性化策略，可以把它叫做个性化推荐，也可以说是“千人千面”，每个人看到的东西都是不一样的，或者说每个人看到的东西都是为他自己量身定制的。想要对用户实现这种个性化的推荐策略，或者是千人千面的内容的分发，首先要知道用户对什么东西感兴趣，要确定知道这个用户的属性。

可以说，准确有效的用户画像是进行一切个性化策略的基础。张瑞说：“我们现在已经初步建立了一套用户画像的体系，并且对一些重要标签进行了挖掘，例如用户的活跃度、People Rank，常用登录地点；作为生产者在特定话题下的权威度；作为消费者对特定话题的兴趣，等等。这些用户标签被用在了个性化排序、推荐、问题路由等一系列任务中，取得了不错的效果。”

张瑞补充说，接下来知乎还会对用户的属性进行更深入的挖掘，例如，对用户进行社群分析，并且定位整个信息传播网络中的关键节点，也就是所谓的 Key Opinion Leader；希望用户的兴趣标签变得更加「动态化」和「可预测」，例如，如果一个用户最近对「怀孕期间的健康」这种话题比较感兴趣，那么可以推测，用户在几个月后，可能会对「育儿」这个话题产生兴趣；他的团队还希望通过用户的分享来重构他的经历，也计划进一步挖掘用户的消费能力和消费意愿，等等。知乎期望能在接下来的一段时间内，能够对用户进行全方位的了解和刻画。

## 内容分析

关于内容分析，为了给用户去做更好的匹配，用户画像从用户方面了解这个用户喜欢看什么。而内容分析，需要知道用户喜欢的这个内容是什么，才能给用户的兴趣更好的匹配内容。

当用户看到比如一个有关于互联网的回答，从用户的角度看可能已经得到了回答，但是对于知乎的团队这还远远不够，张瑞解释说：“我们还需要知道更细一点：我们需要知道他评论的是不是某个技术，或者说某个事件；我们需要知道这个东西，它是不是和某些人，某些地点，或者是某个时间点有关系。然后在什么范围下，哪些人会比较喜欢去看这样的内容，我们会把这个内

容去分享到对应的人的手机屏幕上，或者说他可以接收到这个信息的渠道里面。我们去做这种内容的分析，本身来讲是为了更高效的给用户实现信息分发，或者说是提高他的接收信息的效率。”

知乎上每天都会产生大量的新内容，这些内容需要在第一时间被分析和处理，打上各种各样的标签。因此，知乎构建了一条内容分析的流水线，保证站内每条内容发生变化时，都会立即进入这条流水线进行自动分析，然后第一时间把分析结果同步给搜索、推荐、社区等各个业务场景，对每条内容而言，这个过程大概在 10 秒左右，这种实时性能够满足我们的业务要求。据张瑞介绍，目前在这条流水线上，已经针对文本、图像、音频等数据进行了一些基础分析，例如文本分类、命名实体识别，图像色情、暴恐内容检测，音频降噪等等。

“我们也会逐步往这条流水线上添加更多的组件，例如最近一个重要的工作，是从不同维度刻画内容质量，这些维度包括内容的时效性、专业性、严肃性、准确性，等等。我们还计划对内容进行语义分析，例如自动摘要，让用户在 Feed 流这样信息密集场景中，不用点开卡片就能初步判定内容的大概情况，从而提升筛选内容的效率。”

## 违规信息处理

知乎每天可以产生大量的信息，其中难免会有一些违规信息出现。但是，具体如何处理就成了一个挑战：主要靠人工筛选是行不通的，费事又费力；但是如果使用机器筛选，添加的限制过于严格，就有可能造成误删的情况发生。

信息处理是跟用户体验非常相关的东西，假如误放过一些东西，对于认真讨论问题的用户，或者说是遵守规则的用户来讲是非常不公平的。

“错杀”了之后，对用户的体验造成的影响，也是非常大的。针对违规信息处理，知乎现在整体的思路就是机器辅助运营人员，机器 + 人工的模式去做的整套的一个处理流程。

对于机器来讲，首先是要识别出来信息中的一些不好的东西。对识别出来的东西，并不是进行一个简单的分类，而是有一个基本的置信度的概念在里面。“比如说一段文本，我们可以判断它是一个低俗内容的文本或者是涉及色情的文本的概率有多大。然后在认为执行度非常大的情况下，机器会进入一个自动处理的流程，比如说做一些折叠。这个现在在比如说社区纠纷、用户互相辱骂，或者是色情处理上面，准确度是非常高的，能达到 99% 以上。”如果仍然有误伤的话，张瑞说，用户可以通过申诉或者举报，去把误伤的内容再找回来。

而误伤本身也可以作为下一步模型或者是机器去判别改进的非常重要的样本。加在模型迭代进入的步骤里面对于一些置信度低的处理，需要运营人员做非常专业的判断，去界定它到底是不是一个违规的东西。

## 深度学习

一般谈到机器学习，就会讲到深度学习。深度学习在很多方面，其实都已经显示出了，超越传统的机器学习技术的一些潜力，知乎在深度学习方面，现在也进行了探索，有一些成果。

张瑞介绍说，知乎首先是在基础的内容分析和处理方面，使用了深度学习技术。

例如，用深度学习来识别图片中一些不好的内容，比如违规涉黄的图片。不仅是图像处理，知乎的深度学习还应用到文本分析和处理上面。传统机器学习技术来提取文本摘要，相对有一些

Badcase，这些 Badcase 是制约浅层学习的机制，解决不了问题的需求。而深度学习，在有足够的样本的情况下，能够很好的解决这个问题。

另外在业务上面知乎也在尝试使用机器学习，现在主要用的场景有两个：

- 第一个是推荐

在推荐方面，把问题路由设立一个推荐问题，把问题和用户都打上一些人能看懂的标签，这个标签通过浅层学习，比如说分类聚类的一些方式，去打上的。深度学习的使用不光在问题路由，还包括一些其他个性化推荐，比如说知识市场的推荐上，张瑞说，知乎在尝试了一件事情：通过一个网络，把用户和内容都去做一个嵌入式的表示，这个表示是通过深度学习做出来的，人看不懂它是什么东西，但是机器能够很好的理解。引入深度学习技术，也是因为知乎本身的数据量达到了一定的阈值，或者说达到了一个可以尝试使用深度学习，来去改进效率的水平线之上，所以知乎才会使用深度学习的技术。

- 另外一个使用场景是搜索

搜索里面有很多语义理解相关的东西。如果是像原来自己统计语言处理，或者规则式的语言分析就需要很多的人工去实现。深度学习在收集到大量数据的情况下，恰好能够去做关于用户的判断：比如用户表示什么样的需求，和搜索结果是不是符合。深度学习在处理两个关系之间的匹配方面有比较大的优势，所以知乎尝试把深度学习引入到搜索当中来，顺便做深度的语义匹配。

## 未来的规划

关于知乎未来的发展方向，张瑞表示，依旧会去做用户体验导向的这样一个产品，然后把深度学习技术，或者说把机器学习技术应用在产品上

面，最终目的还是为了去改进用户体验。

在一段时间内，知乎的机器学习技术首先要做的还是支撑现有的产品，同时可能会激发出来跟现在产品方向相关的一些其他产品。关于“其他产品”，张瑞为我们剧透了一些内容：“我们想要做的一个东西希望能够对知乎上面的内容，去进行一个深度的加工。我们现在去做问题路由、做内容的分发，所有的内容，实际上在知乎这边，是经过一个浅层的东西，知乎相当于一个信息的路由器，一边接收一边给发出去但是中间并没有经过比较深的加工。”随后张瑞解释了“深加工”的意思：可以对这些内容进行归纳和总结，比如在用知乎的时候，去搜索一个事件，反馈给用户的，不是说这个事件在知乎上面大家相关的讨论都是什么，而是总结一下，关于这个事件大家在知乎上面讨论的热点都是什么。在这些热点上，有多少人表现出了不同的观点和看法，有多少人

赞同、有多少人反对。这是提高用户接收信息，或者说接收知识效率的非常重要的一个改进。“我们在尝试去催生出一些这样让用户使用起来更方便的产品。”

张瑞随后补充道，另一个设想是期待机器学习技术能够帮助知乎构建一个更加智能、高效的用户交互接口。“一个设想是：智能问答产品，使用自然语言生成手段，以一种更加自然、易于理解的方式，将知识库中的信息提供给用户；更进一步，我们是不是可以让知乎成为一个汇聚了众多用户的知识和见解的大脑，和用户进行更自然、形式更加丰富的相互交流呢？这也是业内的一个比较前沿的研究方向，像 MSRA 和 Google 研究院也在做类似的工作，我们也希望知乎能够在这方面能有自己的积累。”

**张瑞**，知乎机器学习团队负责人。2012 年毕业于北京邮电大学通信工程研究生专业，毕业后一直从事搜索引擎及自然语言处理方向的研发工作，此前曾就职于百度及豌豆荚。

# 回到网易8个月来的测试团队转型实践

作者 李乐



2016年初月回到网易，进入交友事业部，更加专注于移动互联网APP研发测试领域，在将近一年来的时间里，经历了开发、测试团队的转型，下面讲述带领测试团队从挖掘痛点的转型实践。

## 测试团队现状

交友事业部人员朝气蓬勃，个人认为更像一个创业型的公司，初期技术资源都投入到产品功能需求开发中，对于产品质量稍作妥协，不需要太严格的过程控制和质量把控，相比开发资源而言，测试的投入资源不是那么急需。

随着用户量的上升，各种类型的移动设备问题错综复杂，用户对产品的质量有要求，部门

老大对质量越来越重视，狠抓这块，从2015年Q4、2016年Q1分别招入两名测试人员，整个技术团队对于质量把控的诉求越来越强烈了，到后来整个测试团队跟随开发团队的规模壮大而壮大起来了。

## 开发测试人员配比

交友事业部有三款APP产品：同城约会、美聊、花田，一线开发人员总数20人，一线测试人员总数4人，示例如图1（2016年Q1统计）。

图中可见测试开发比例是1:6，Android、iOS端各占一名黑盒测试人员，后端API无相关测试人员参与。

	按端分	开发人员	测试人员
美聊	后端	4	0
	Android	4	1
	iOS	4	1

图1 开发测试人员配比

## 测试技能现状

所有产品线的测试手段都是以手工测试为主，无自动化辅助手段，回归测试成本高，Android、iOS 独占测试人员忙于业务的功能性需求的黑盒测试，非功能性需求无法满足。

Android、iOS 与后端 Server 进行数据交互的 API 规范定义是一致的，早期无相关测试人员参与，导致发现 API 问题较晚，推迟到客户端功能开发完成阶段才进行检验，同时也造成后端 API 回归成本高；

功能测试以及 API 相关测试在研发测试过程走一轮、预发布环境第二轮、生产环境走第三轮，深度依赖于手工测试，发现问题滞后，相比需求、研发阶段修复的成本来说，发现的阶段越晚修复成本越高，最终可能导致带着严重问题上线运营。

## 测试流程现状

交付式测试，开发人员把相关功能任务设置为 done 之后交付给测试人员，测试人员未全程参与从需求源头开始跟进（及时了解需求背景和细节，消除需求含混性，及早开展测试用例编写工作），从而研发过程中客户端功能、后端 API 的可测试性（一个完整的功能是可以分多个功能

小点提测，最终完整再提测一次）无法提高，测试人员也无法及早进行冒烟测试；

无测试人员专属的持续集成构建环境，Android、iOS 打包依赖开发，测试人员存在时间等待上的开销成本一直存在未能降低。

测试三轮验证：测试环境验证第一次、预发布验证第二次、生产验证第三次，为什么做三轮，这三轮的评估依据是什么？

整个测试过程，只有测试人员参与，产品、客户端开发同学的协助如何提升融入进来呢？

## 测试任务评估没有依据

针对需求的相关测试任务，出牌评估工时，没有评估依据，直接拍脑袋进行，体现在：这个需求需要测试哪些方面？涉及客户端 Android、iOS 哪些特性？有哪些兼容性需要测试？只有把所有相关点列出来，评估完整的时间，再进行合理的取舍，让质量维度维持在一个可接受的平衡点，而不是一味追求最高质量，往往很多时候，利用现有资源做最平衡的质量优化，可接受的容忍度。

所谓平衡点的简单例子：

- 字体样式的问题，并非致命的，可以权衡接受跟着上线；
- 客户端列表过长溢出，没有边界判断机制，这就是致命的，必须修复上线；
- 客户端数据出错了，后端还可以通过快速发布来解决，并不影响客户端的上线。

## 生产力改进实践

生产力改进实践环节，是围绕几个大方面开展的，如图 2 所示。



图2 生产力改造围绕方面

## 敏捷开发

建立 Scrum 流程框架（版本开发流程），以此为基础的版本开发模式，各个角色紧密配合的 PDCA 循环：高度合作，善于计划和总结、拥抱变化、高度可视化，如图 3 所示。

## 自研的燃尽图进度跟踪工具

过去 Jira 任务管理系统自带燃尽图不能根据团队特点，展示实际进度和体现反馈风险所在，导致错过反馈进度问题的最佳时间，因此根据团队特性，自研能够反馈实际进度的燃尽图，让项目进度透明化，技术、视觉、交互、产品都参与到风险识别和反馈中来。

带来的效益：

- 使用新版燃尽图之后，每日晨会分析历史进度问题有依据，能够明显看出风险所在；
- 产品人员主动关注燃尽图趋势变化，及时调整有问题的任务，提高研发交付的时效；
- 每日工时可以看到研发、测试人员的个人进度，及时沟通遇到的困难，推进解决。

负责客户端的测试人员承担产品职责单一，

技术要求多层次

最初测试人力资源不足，为了提高更大的复用率，要求每位测试人员负责客户端 Android、iOS 的两端的测试工作，编写一份基础用例，根据每端特性在测试过程中再改变策略，落地实施的第一个季度就暴露出问题：

同时兼顾一个产品多个功能的测试任务，对于客户端开发同学而言，他们是并行工作的，而测试同学需要在不同功能的 Android、iOS 两端来回切换，导致效率低；

同样问题也存在兼顾多个产品的测试任务，有些产品是同时进行的，需要在多个产品的任务中切换，导致对两个产品都不熟悉；

测试设备占用时间严重，在进行 Android、iOS 轮换切换的场景中，一人独占相关设备；

改进：单一职责，专职专责，原则上不再进行跨项目的版本任务，也不在版本中负责一个功能的 Android、iOS 相关测试任务（除了运营的相关活动项目可以兼顾 Android、iOS 测试），主攻 Android、iOS 单一方向的功能测试、自动化测试，说的高大上一点好像成了全栈测试工程师。

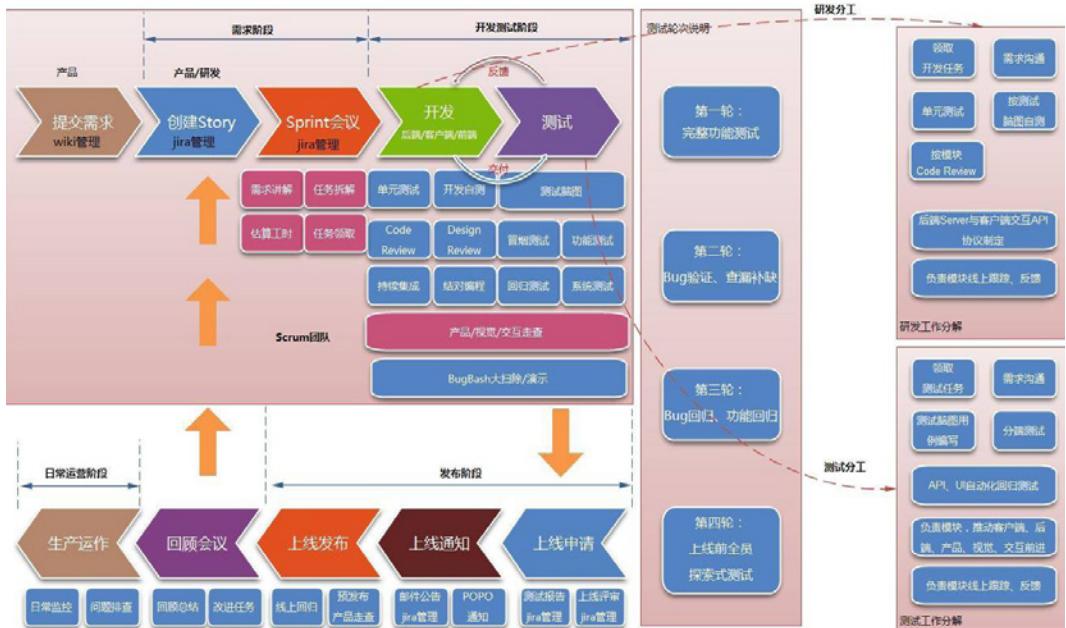


图3 Scrum流程框架-交友

实施半年之后，收益颇深，各自负责Android、iOS的测试同学结对编写测试用例，抽取共性部分，运行时附加个性化的系统特性，并行测试效率提高，设备占用率降低。

## 自研的API管理和测试平台

过去后端的API规范是通过word文档进行管理，版本变更需要手动通知相应人员，而且每个人编写的格式不统一，容易造成冲突，解决上有时间开销，另外修改跟踪反馈上的成本很高，开源项目中也没有能够适合交友团队模式的工具，因此投入开发API管理和测试平台。

考虑到客户端与后端交互是通过API进行，将API平台化管理带来效益：

- 使用平台化管理清晰呈现Mobile API接口分布图，有效减轻了后端同学管理接口规范的

工作；

- 方便客户端同学快速查阅和版本对比；
- API修改历史记录对比，修改后第一时间系统自动通知相关人员；
- 在接口定义完之后，可直接生成API Mock，节约手工写mock接口的时间，客户端同学可以直接开始开发工作，与后端开发并行。

功能点包括以下三个方面。

## API 统一规范

支持在线管理接口规范文档：接口规范管理功能有很多特性，包括自动生成change log，自动生成技术审查的规范文档，review通知，接口版本管理，支持任意历史版本的对比，方便追踪每个版本的变化。

后端同学只需要专注于接口定义，大大节约了文档维护的时间，更早投入开发工作。

## API 模拟调试

平台支持从接口规范文档直接生成 API Mock，在后端接口开发完成之前，前端、客户端的同学利用 Mock Server 摆脱后端接口的依赖，直接开始开发工作，与后端开发并行（见图 4）。

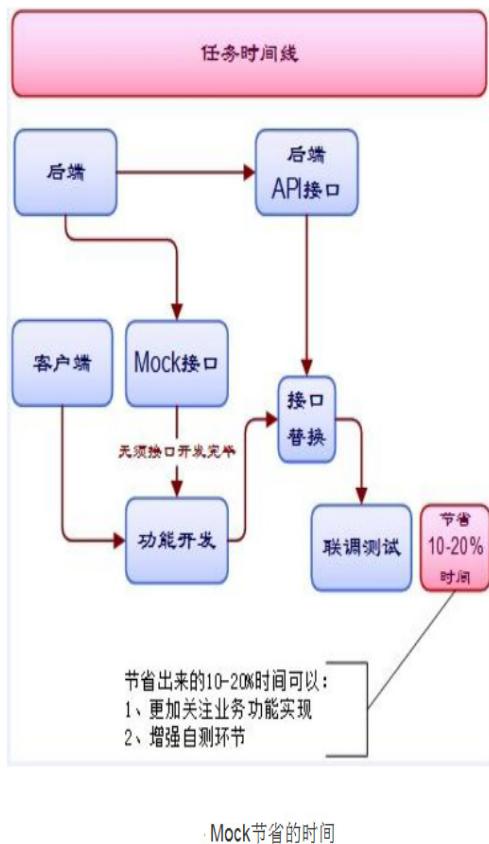


图4 API模拟调试节省时间

## API 自动化测试

平台支持从接口规范文档直接生成 API 测试用例，测试人员集中参与关键场景编写，执行用例之后自动生成测试报告咯，测试同学可以在后端开发的同时，写好测试用例，在开发完成后做

冒烟测试，以及回归测试，提升测试效率（见图 5）。

## 持续集成与静态代码分析

过去代码构建在开发人员本地进行，每次提交在解决冲突上时间开销大，每个环节发现的问题滞后，无法自动化集成、按需构建，以及代码的质量没有数据参考。

团队需要引入有效的自动化构建平台，以及静态代码分析平台，用以指导日常开发过程的质量改进，将代码问题的反馈机制自动化，构建数据可视化。

## 持续集成

为了让产品可以快速迭代，同时还能保持高质量。技术团队对各产品的各端都建立了持续构建平台：在代码集成到主干之前，必须通过自动化测试。只要有一个测试用例失败，就不能集成。保证持续地发现、反馈和解决问题。

## 静态代码分析

为了保证代码质量，从代码层级降低线上出错的可能性，技术团队引入了静态代码分析技术：在不执行计算机程序的条件下，对源代码进行分析，找出代码的设计缺陷，例如代码规范、内存泄露，以及体现总体质量：代码覆盖度、技术债务的趋势图，通知技术改进，拦截在上线之前，这些数据都成为 QA 统计的数据来源。

## 客户端手工覆盖度数据收集工具

过去执行完测试用例之后，无法考量哪些代码覆盖了，哪些没有覆盖，测试用例写的好不好，为了解决这些困境，在客户端 Android、iOS 植入手工测试覆盖度工具，收集代码覆盖度展示，

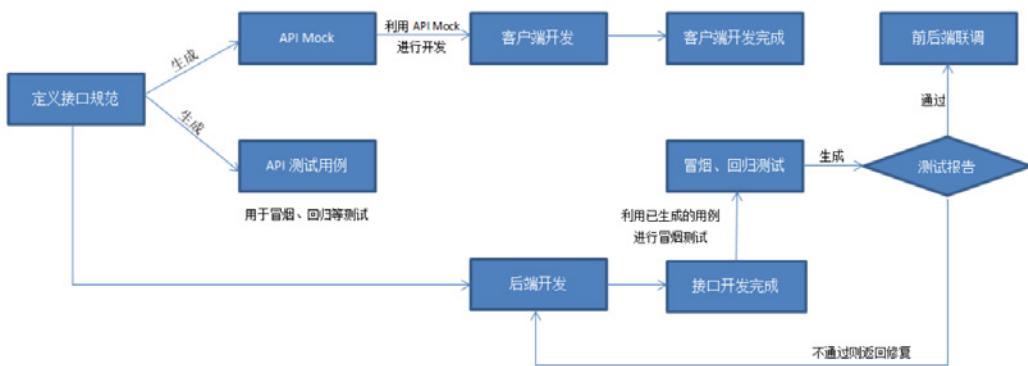


图5 API自动化用例流程

目的是找出测试过程中未被覆盖的代码，指导测试人员调整测试策略，开展探索式测试。

图6是执行美聊2.8版本iOS相关用例后的统计结果，可以根据结果调整测试策略，例如：如果改动了登录模块，目前用例覆盖度比较低，那是需要加强特殊场景测试，还是其他方面呢？这个需要团队review下做出决定。

## 利用BugTags工具的问题反馈

过去发现线上问题无有效收集数据的手段，用户反馈之后，需要相关人员跟进沟通，询问环境、设备等诸多问题，整个过程繁琐，人力投入开销大，引入BugTags是为了简化Bug提交过程，记录重现场景相关信息，将客户端的大量复杂操作最大限度简化。通过白名单机制，美聊可以让用户打开Bugtags摇一摇问题，提交用户的相关环境、设备信息，进一步推进排查问题的效率。

## BugBash质量活动

传统的产品走查，产品、视觉、交付、运营只对自己负责的功能部分有了解和检查，缺乏一个需求方的整体走查。当有人发现一些功能间互

美聊iOS-2.8.0版本覆盖率数据见下表

2.8.0覆盖率数据详情脑图

模块	Line Coverage	Function Coverage
登录	32%	42%
主页推荐	42%	37%
搜索	63%	65%
聊天	67%	72%
悬赏	62%	65%
小圈	85%	90%
emoji表情	84%	90%
碰缘分	66%	80%

图6 美聊2.8-iOS手工覆盖率

相关联的问题时，已经比较晚，修复成本高。引入Bug Bash（所谓Bug大扫除的活动），在项目开发阶段的末期，专门划出一个专门的时间段（通常1天），打破以往非技术人员未参与的做法，在这期间所有参与项目的人员（技术、产品、交



图7 Bugbash流程

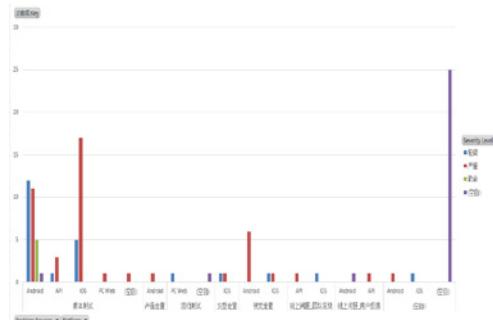
互），集中全部精力，运用各方面的知识来搜寻项目的 Bug，做到及早发现问题。

会后将问题汇总，用以推动开发改进功能（见图 7）。

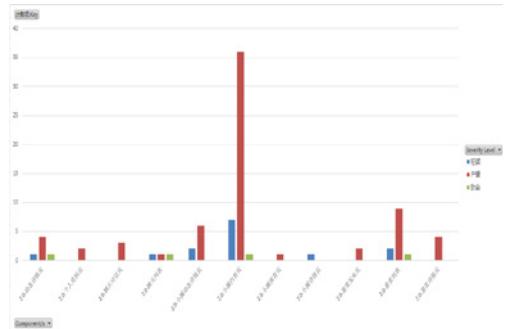
QA数据收集

在 Sprint 总结会上为了让项目成员能更加清楚了解整个 Sprint 的质量、进度问题，从 Q4 开始对每个 Sprint 都做了数据收集和展示。通过收集每个迭代版本的工时、bug 数据，在总结会上向全体人员（技术、产品、视觉、交互、运营）呈现当前版本总体质量多维度数据，指导工作的改进方向。

- 按照阶段的bug分布展示



- 按照组件的bug分布展示



## Android Monkey崩溃性测试

持续集成环境每日代码 daily build 之后，  
夜间在测试专属服务器进行长达几个小时的  
Android Monkey 崩溃性测试。

## Summary

- Hardware:XT1096,TA46800H9W
  - Application:com.netease.huatian
  - Span:2017-03-16 19:20:15~2017-03-17 04:44:01
  - Results:0,Crash

## Details

No	Result	Duration	Event Count
1	OK	9h 23m 45s	13800

图8 Android Monkey崩溃性持续构建

## 兼容性质量风险控制转移

目前交友测试团队现有的 Android 测试机型不足，为了解决 Android 碎片化，特别是兼容性

问题，借助公司内部的易测平台来控制质量风险。

重点关注基础兼容性：安装、启动、monkey  
随机、卸载。

## 团队人才建设

16年初的测试团队规模太小了，业务测试需求不足以满足，人员技能集中在黑盒测试，没有移动UI自动化测试、后端Server API自动化测试、测试平台开发的相关经验，并且全员对于Android、iOS代码不了解，白盒测试无实践经验，也会导致排查问题不够深入了解原理。

从16年Q2开始制定团队建设技术，那么整个测试团队的关注点是什么，如何聚焦，根据技术总体需求、产品需求来落实测试需求呢？

根据团队特性，测试、开发划分了边界，只有从这些方面出发，才能更好要求组员的技能形成阶梯化，以及在招聘要求是按照此需求来落地，市场上大有可为之人，如何切实际为之更重要，下面从几个方面来谈谈。

## 测试团队关注点

Martin Fowler在博客中解释了TestPyramid，如图9所示。

单元测试是第一道测试关卡，也是一个陷阱，测试人员如果投入到此环节上，将是一种资源耗尽型的质量活动。比业务熟悉程度，测试人员没有开发人员高深，比写单元测试的效率，测试人员没有开发人员高效，这里交友测试团队也跳坑了，历经一个季度跳入、跳出，理想的状态下是：开发的框架很松耦合，例如使用了MVP/MVVM开发模式，实际情况是这些技术债务在逐步偿还，熟悉代码的开发人员进行单元测试都有阻碍，测试人员谈何容易，简单点来说不务正业，投入产

出比低。

真正要从业务需求的痛点出发挖掘适合团队的方向：测试层次的关注点是最清晰的一条分水岭隔离开代码级别的：单元测试、集成测试，测试人员真正的关注点是：以手工测试为主，自动化为辅的发展阶段，同时围绕整个研发测试过程的质量反馈，包括：需求阶段、开发阶段、发布阶段、运营阶段。

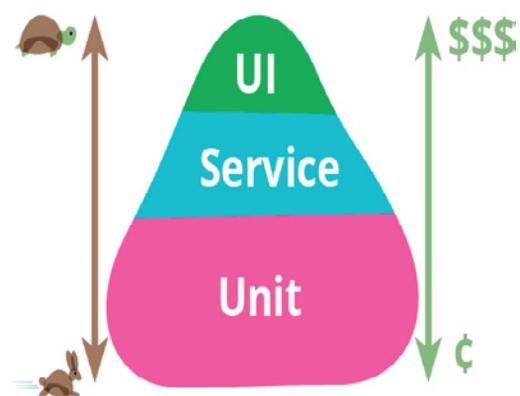


图9 Martin Fowler: TestPyramid

理清整个需求之后，就是团队成员角色转型（见图10）。

基本职能：手工测试工程师，进阶职能：自动化测试工程师，再高级一点，测试开发工程师，其实也可以称为全栈，名字不是最重要，也不会设立这种title，只是要明确把活给细分出来。

最后，根据需求，也把产品测试人员分布明细理顺了，见图11。

按照此规划来落地招聘需求，避免因人设岗，而是实实在在的产品需求、技术需求来决定人才所向。

## 测试团队文化建设

由于篇幅有限，简单来说形成学习分享的技



图10 岗位的转变

产品测试人员分布明细

	手工测试	自动化用例编写
美聊（投入4人）	手工测试 iOS 0.5人（兼任） 1人 Android&iOS运营活动 0.5人（兼任） 0.5人（兼任） Android 1人	自动化用例编写 iOS 0.5人（兼任） API 0.5人（兼任） Android 0.5人（兼任）
花田（投入3人）	手工测试 iOS 0.5人（兼任） 1人 主站&运营活动 0.5人（兼任） Android 0.5人（兼任）	自动化用例编写 Android 0.5人（兼任） API 0.5人（兼任）
自动化测试平台（投入1人）	1人（兼任）	

图11 测试人员产品线分布明细（2016年Q3）

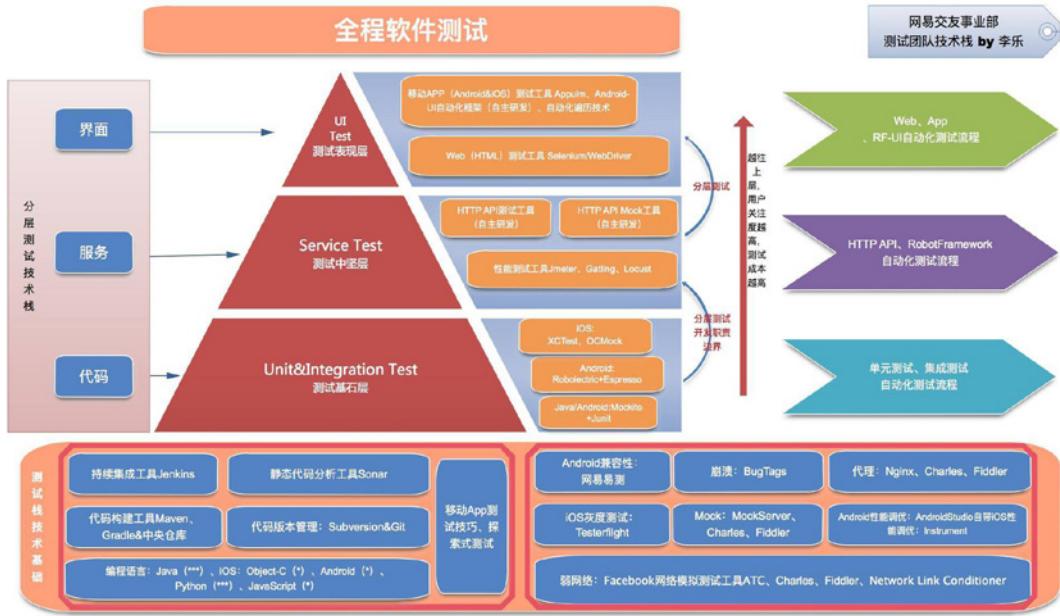


图12 测试人员产品线分布明细（2016年Q3）

术氛围，让测试人员定期组织技术分享，这些技术主要是可以用于生产落地以及对新技术的调研成果展示均可，另外有一些虚拟组设置，例如：自动化测试组、平台开发组，用于把兴趣相同的组员融合到一起，投入到合适的方向上。

以上是本人在网易交友事业部一年以来对测试团队转型带来的分享，在合适的阶段对测试资源做合理的投入是有必要的，发展初期的困难适当取舍产品质量，换来更多功能亮点吸引用户，占领市场，站稳脚步，发展中期，确保用户的活跃、稳定，是需要靠产品质量取胜的，产品功能并不在于多花俏，有新意、简单化、易传播这几个点可以适当考虑，其实到了中后期，技术很多处于还债阶段，之前设计的系统业务模块解耦、微服务化，提高可测试性都非常重要，而测试人

员往往对于技术还债的重构要更加留意，一不小心就掉进坑里，久久不能自拔，同时最后牺牲最宝贵的就是测试质量，这是需要取舍的，别以为质量就是高高在上，测试团队的利益应当与开发、产品团队的保持一致，这才是发展的硬道理。

另外，在接下来一年有计划的话，交友测试团队会把关键环节的实践在 InfoQ 逐一分享给大家，敬请关注，最后附上一张《网易交友事业部测试团队技术栈》（见图 12）。

# 如何成为一个合格的技术 Leader?

作者 林日华



## 编者按

在 10 月份上海 QCon 大会上，百度外卖研发中心总监张灿将带来演讲《向前一步——年轻技术管理者的涅槃重生》，InfoQ 在此之前，对张灿老师进行了一次独家专访，让张灿老师聊聊作为女性技术人的成长感悟与对技术人转向管理者的思考。本文即由采访内容整理而成。

大家好，我是张灿，现在就职于百度外卖，目前我负责的是百度外卖开放物流相关的一些工作。今天给大家分享一下我作为女性技术人在 IT 行业中的成长感悟，同时讲讲我关于技术人转向管理者的一些思考。

说到女性技术人，大家马上会有“人数很少”、“在这个行业里路途艰难”之类的印象。但是女性程序员其实在整个行业里比例不算太低，行业统计应该是 20% 左右，这个比例我认为没有大

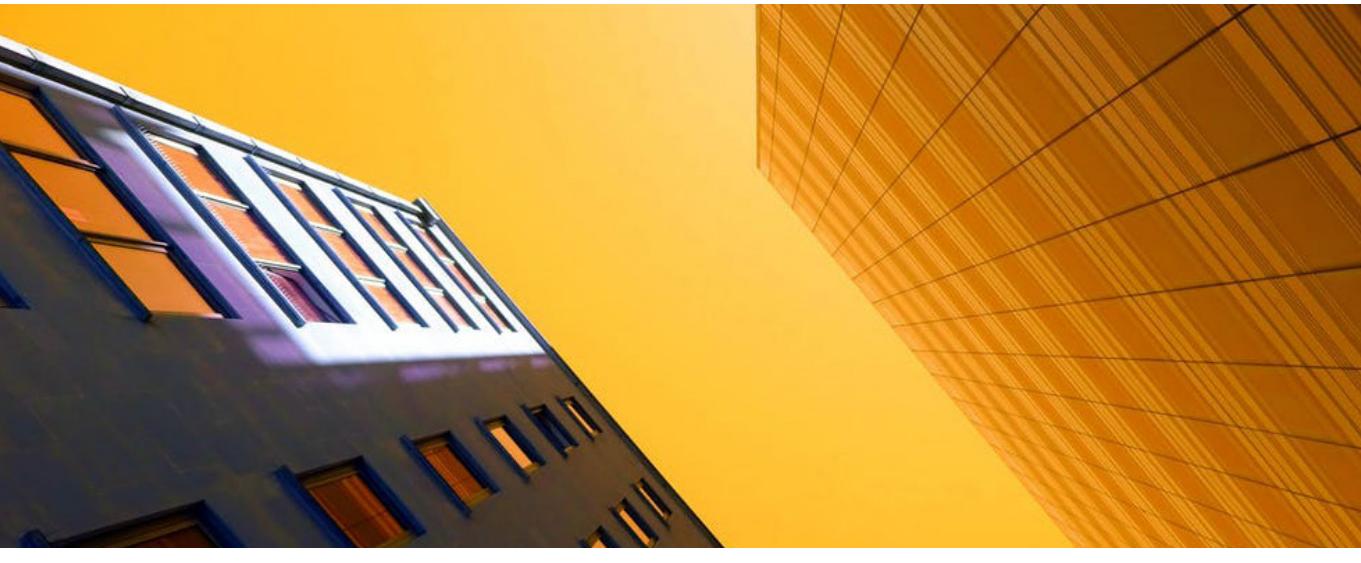
家想象的低，至少我身边的一些女性管理者或者是女性 coder 还是蛮多的。

然后辛苦这个词我觉得谈不上吧。我重点给大家简单分享一下我是怎么踏入这个行业以及在这个行业里面我认为遇到的一些比较大的方面的问题和一些困扰。

## 女性技术人的困难

因为我本身大学就是学习计算机专业的，所以我没有觉得说最开始找研发类工作特别地困难，我觉得这还是一个蛮自然的过程。09 年我加入到百度的时候也是蛮机缘巧合的一件事情，因为当时我还是在校的大三学生，也是因为一个师兄的推荐，所以就在暑期参加了百度的那个实习招聘，然后就来到百度。

当时我原本的一个职业规划是想去国外读研



究生的，但是也因为这段时期的经历，基本上整体改变了我自己的一个职业轨迹。当时我自己所在的团队的伙伴都非常 nice，而且我对开发这个事情还蛮有兴趣的，所以最终我 11 年毕业的时候就整体加入了百度。

我觉得这一个过程还算是蛮顺其自然的，然后参与开发其实也没有很困难，我觉得“痛并快乐着”这个还蛮贴切的。接下来我给大家谈一谈我自己的一些关于“困难”的点的真实感受。

## 自信心

首先，就我见到的情况以及我个人的经历来看，我觉得在 IT 这个行业里，女性普遍不太自信。本身这个行业是男性主导的，80% 都是男性，而很多女性从业者可能会习惯地低估自己的能力与工作中的表现。

关于这个，在我曾经还是一个一线工程师的时候，我深有体会。包括我自己，其实也曾经有过一些困扰，比如说同样一件事情，可能我的确在团队里是最专业的，但是我又很难非常自信地站出来把这个事实说出来。整体来讲的话我觉得一个女孩子如果希望在这个行业里面有长足的发展，那么需要去不断地提升自信心。

在我做管理之后，我对我的团队里的程序员做过一次小小的测试，当时我让他们简单地自评一下个人的工作。我印象很深刻，有一个女孩子其实在某一个领域里面，她非常专业，可以说是当时我的团队里面的一把手，但是她给自己的打分还蛮保守的。并且这不是个别现象，其实整个团队里的女性程序员，最后的整体自评分都比男性低。

总结一下，我觉得自信是挺关键的，一路走

来，我觉得对自信心的培养，就是需要你 坚信自己确实可以做得很。

## 团队认可

第二个层面来讲的话是 团队认可，女孩子想在这个行业里有一个比较好的团队认可度，其实是要付出可能比男孩子更多的努力，甚至还要接受更多的质疑与挑战。我想给大家一点建议，这只是我一路走来个人的一些感受。如果你希望自己在这个行业里能够跟男性 coder 平起平坐，甚至比他们表现得更优异的话，就接受这样的现实，并且努力往前走。

## 职业迷茫

最后一点我觉得是这个行业的一个通病，就是 职业迷茫。这个其实与男女无关了，就是大家都会有这样的问题，但是女性会表现得更突出一点，因为女孩子可能二十八九岁会面临比如说照顾家庭啊之类的一些思考，然后要经历一些权衡，所以可能会更容易出现这种职业迷茫，会出现“我是不是要继续做这个行业？”“我要不要转型？”这样的一些职业迷茫。我觉得这个对女程序员其实还蛮煎熬的。

回过头来看，其实我也有这样的职业迷茫期，我 坚持 过来了，没有什么金科玉言给大家，但是就是要 坚持。我觉得如果对这个事情足够有兴趣，而且认为自己还蛮有能力把这个事情做得很好，认为在这个行业里还有自己的价值和位置，那么在职业迷茫期里，你就去坚持一下，之后就会有一些新的职业阶段去展开。

## 转型成管理者初期的困难

我最开始技术转管理就是在百度，刚开始转型的时候其实每天压力都非常大。那个时候很苦恼，因为基本上就是一个问题接着一个问题来。在做技术的时候可能只需要专注地去做好一件事情或者去突破一个问题就好了，但是管理因为它涉及到人与合作，比如说要跟进相关问题，要去促使整个业务的一些落地执行等等。另一方面比如说团队成员状态不好，其实这些都是你要处理的事情。

我觉得最初期的时候我还挺疲于应付这样的问题，如果回过头来去总结的话，我觉得主要是两大层面。

## 思路的转变

思路的转变，这个是我认为初期还蛮困难的一件事情。我在做工程师的时候，个人的这种战斗力还是蛮强的，在团队里各方面的问题可能都能解决得很好，但是带团队的话你是需要去做一个转变的。你需要把自己的个人战斗力去转化成团队的战斗力。

那个时候的我就觉得“这就是一句话的事啊，不就是个人到团体战斗力的一个转型吗？”，但是在实际工作过程中，才体会到，很多事情你是需要很刻意去练习的。比如我自己需要组织团队里的其他人一块去做这个事情或者说把这个事情做得很好，这种组织之类的事情，需要我去实践、去练习。我觉得很多管理者初期都会碰到的这样的问题。

## 执行

核心的一点应该是如何实际地去解决一个问

题，就是对工作的 执行。作为管理者，首先，你需要去把任务做拆解，让团队的成员一起参与进来，并且去执行。

我觉得这个事蛮困难的，因为在以前我只需要技术专业就好，但是带团队的话，除了技术专业，还要有足够广的 知识，或者是 认知 和 推理 这样的能力。这样才能将一个任务从提出，到拆解，到分配，到执行，再到反馈等一系列的过程给处理好。你需要有多元化的手段去解决这个任务在执行过程中遇到的问题，这就需要你有足够的多方面的能力。

## 新团队的组建

我真正意义上组建第一个团队应该是在百度外卖，当时的研发部门业务发展非常迅速。O2O 外卖刚刚兴起的那段时期，我们团队基本上就是零规模，只有 2 个人去做这样的一个事情。然后当时整个业务的扩张速度非常快，这意味着我的技术团队必须要跟上。

那个时候我也是临危受命吧，也就是在这样一个机会下去计划组建一个研发团队，并且快速地去解决当下业务的一些问题。这里也是从几个层面来谈谈组建团队的一些问感想吧。

## 学习

当时我做了比较多的一些基础知识的了解，这应该是我当时组建团队的第一步。这个基础的了解其实具体到了包括说我要去了解整个 现有业务 到底面临什么样的一些问题。这个就真的是一张纸一点点去写啊。

然后就是这个团队未来其实还是要在外卖这样一个大的环境下或者说在百度整体的大环境下发展。不仅要学习外卖业务的一些东西，我也

要学习百度对于管理者的一些要求，或者对团队这样考核的一些机制。

可能以前我们作为一线工程师，只需要被动地接受考核就好了，但是当我站在了这个团队领导这样一个角度上，其实我是需要很宏观地了解这个考核机制到底是怎么运作的，这些都是需要去学习的。

## 团队规划

通过这样的一些基础学习之后，最终就要生成一个叫做 团队规划 的东西。团队规划其实包含的元素很广，比如说 团队的规模、团队短期的目标、团队要解决的问题，以及 团队分工 等等。核心来讲的话应该就是从 业务 和 技术 这两个大的方向去做一些拆解。

具体来讲就是基于这样的一个层面的规划去做整个人员的分配，把合适的人分配到对应的位置上去。我最开始的时候其实团队只有三四个人，但是当时要解决业务问题，我至少要把团队扩张到 15 个人左右才可以，没有人的话就意味着我完不成这个业务，所以没人，那么分配也就无人说起了。

这时候就要考虑招人了。人手问题，这是现在很多团队可能都会面临的一个困境，所以对于管理者来讲，这时候就要更多元化地去解决这个问题。当时我们没有办法，招聘不到合适的人，那就通过用技术化的一些手段去规避人手这个问题，比如说我们的技术选型是不是可以折中？我们是不是可以用其他的工种去完成这个工作？

我觉得要看一个具体的业务情况以及你要组建的团队，具体要干什么，这个需要详细地做一些分析。总体来讲就是 多元化，动用一切你能够使用的资源，去把你团队规划里需要匹配的

角色以及这个角色的空间要求之类的东西拿到。

## 文化建设

文化建设这一点应该是在你的团队初具规模的时候就需要去重点培养和关注的。因为你去建设一个团队，如果这个团队没有一些隐性的文化，其实很难有很强的凝聚和归属感。管理上称为“团队氛围”，就是如果没有这样的话，大家在这个环境里面可能都不会待长久。

### 不断优化

之后就要逐步地去做一些优化了。因为业务发展阶段不同，我们所需要的团队梯队其实也是会发生很大的变化的。这个在互联网企业里面可能会更常见，可能两三年这样的一个时间里，你的组织架构就会发生重大的变化，需要根据业务发展来不断地对自己团队的各方面进行优化。

## 关键的点

我觉得对于 leader 来说，有个关键的点需要注意。就是你要做一些 先于团队成员的思考。我之前给初级管理人员做培训的时候讲过，哪怕拿出一两个月的时间去想一想，自己希望团队未来是什么样子的？

这个的话不仅仅是站在你当下所处的业务范围内，甚至要站在这个行业里面去思考。比如说你带的是一个基础架构的团队，那么你就要去对标，思考一下其他业务的技术架构团队都在做什么，以及你的团队能不能做成这个样子。

用宏观的角度来讲，就是说你的团队在业界需要有一些什么样的影响力。我觉得这个是需要 leader 先于团队成员去思考跟运作的，这个对于整个团队的空间和技术良性地沉淀其实有很长远的意义。

## 年轻女性管理者的优势与劣势

我从我自身的角度给大家分享一下，通过我自己的分析还有跟我比较相近的一些人他们在做就这种日常工作时的情况，来看看在管理上年轻的女性管理者到底有什么样的优势和劣势。

### 优势

先说说优势吧。

- 执行力

周围的人对我的一些评价是 执行力 非常强，因为可能年轻人比较有冲劲，比较有魄力，也比较有激情，去做一件事情的时候，就会比相对年长沉稳一点的人，显得整个执行力很突出。这应该是年轻管理者的一个很强的优势，这也是为什么现在很多公司会愿意启用年轻的管理者，愿意给年轻人机会的原因。

- 韧性

我觉得从女性的角度来看的话，一般女性管理者的 韧性 还是比较强的。举个例子，之前我们在与对外的部门合作，比如说跟百度什么大数据啊之类的部门做合作的时候，我会明显感觉女性领导在整个会议中的表现更有耐心，而男性领导会急躁一点。当然这个也跟个人的性格有关，但是我觉得这的确是男女的差别，科学研究结果也显示其实这是一种天生差异。

- 沟通协调能力

其次，大家都会评论说女性管理者的 沟通协调能力 比较好。我觉得沟通协调能力对于一个管理者来说是非常重要的。对于任何一个管理者，在踏入这样的一个领域的时候，沟通协调能力都是非常核心的主题。

我印象很深刻，百度当时对于初级管理的一些培训上有一个非常重的课程，就是在讲跨部

门怎么沟通，对内向下怎么去做沟通之类的。

我觉得一般女性情感与心思会比较细腻一点，所以在整个沟通上面可能会更照顾到团队成员的感受，那么在整个目标的传递上可能会更能准确地进行。这一点我觉得是天然优势。

#### • 亲和力

还有一个是 亲和力。亲和力可能是团队中很关键的东西，这个我有比较深的体会。我团队里有 90 后的年轻人跟我聊天的时候，说觉得跟我聊天没有什么负担，因为他之前在一个平均年龄 40 多岁的公司，他觉得公司氛围非常压抑，他工作干得不开心，很影响发展。

然后在我的团队里，因为大家年轻，思想也比较有活力，可以更多地去交流，在一个相对轻松亲和的环境下相处，所以工作也会变得积极有进度。

总之我觉得女性在沟通、激励以及亲和力之类的物质上，其实优势还蛮明显的。

## 劣势

但是劣势也比较明显。

#### • 阅历

我觉得首先就是因为年轻，所以在业务等各方面基本上就比较薄弱。我观察到跟我处在同样职业阶段的一些经理人，可能很多人都是在这个行业里做了十年甚至更久了，所以就不得不承认这是有一个客观的 阅历积累 差距在。

在这个层面上我的处理方式就是 虚心学习。我觉得去跟他们聊事情或者说跟这样的一些长辈去达成共识的话，更需要一些低姿态的方式，因为你还年轻，还有很多东西需要学习。

#### • 情绪化

第二点是，我们说女性管理者还是有一些偏情绪化的。前面我们也提到了女性在整个的这

种沟通协调方面还是比较好的，这点其实我觉得既是优劣也是劣势。从劣势的角度来讲的话，这种情绪化有时候会显得不那么理性，会把事情往一个不那么合理的方向去带。

#### • 威信

然后第三点，是一个核心的问题：威信的建立。因为我本身比较年轻，我的团队当时六七人规模的时候其实有很大一部分比例的成员是比我年长的，我觉得他们的工作年限与资历都比我高。这个有点天然的感觉，但是在这个团队里，他们就是需要汇报给我的。

那么在团队中，我的威信的建立就需要去多花一点心思。我觉得这一点在 10 月份上海 QCon 大会上的分享里可以重点给大家讲一讲。

## 大环境

最后一点，我觉得是 大环境劣势。大环境劣势就是说整个行业，其实对于女性从业人员，多多少少还是有一些偏见和一些问题的。

感触比较深的一个 case 是，当时我们有一个女高管，她在自己的办公桌上摆了一张自己跟小孩和爱人的合照，大家看了这样一张照片后，本能的反应是想“这个女高管是不是在自己家庭上的经历会比较多呀？”

但是如果是一个男高管放了这样一张照片，那大家会觉得这个男人非常顾家之类的。这种细微的差异对待其实就反映了所谓的大环境劣势。

我给出的建议是心态放平就好了。平和一点、坦然一点去面对这些问题，并且要勇于承担起这个大环境带给女性的一些问题，同时学会聪明地去化解。

## 给新晋管理者的建议

我更系统地给新晋管理者一些建议吧。

- 最开始的时候，你可能要学会 管自己；
- 接着要学会 管别人，其实也就是 管团队；
- 然后再接着就是学会 管组织，管组织是指 团队规模发展大了，就与“管团队”有所区别。从管理学上来讲的话，一个人能够直线管理的人应该就是 6 到 12 个，如果超过这个规模的话，其实你是没有办法的，所以 就有一个“管组织”的说法；
- 最后一点是 当领袖，这个比较遥远，也 比较宏观了，现在就不谈了。

## 服务的心态

今天讲初级管理，那么我觉得第一点与第二点可能会更关键一点。我给大家一些建议吧。首先是 服务的心态，我前面有提到，在管理初期其实会有一个很大的困难，就是个人作战能力向 团队作战能力转变的困难，现在回头来看，我觉得核心的瓶颈就在于我的服务心态。

因为以前一个人的时候不用管别人，我干好自己的事情就好了，但是现在我要去让别人上，我要给他一些机会和空间，甚至要帮他去完成一个个问题。总之我就是需要给出建议与帮助，我觉得这个是蛮大的一个挑战，重点就在于服务的心态要先摆正了。

## 协同合作

第二点我觉得是 协同合作。对于初级管理者，你需要去把自己的团队中不同能力、不同性格、不同风格的人打包在一起去工作，这个事情其实听起来很简单，但是在实际运作的时候，是

一个非常困难的事情。如果没有冲突就还好，一旦两个人的发展或者是成长的空间有冲突的时候，就看管理者怎么去做，以什么样的方式去处理了，这还蛮考验水平的。

## 规章制度

最后一点是 规章制度。如果没有规章制度去管理，那么你自己的管理其实是会吃掉你大量的精力，所以去建立整个团队的标准其实非常关键的。

我们说团队氛围、团队形象，其实都是团队文化，其实也是规章制度的一种。这就需要你规矩去办事情，这个规矩不一定是说你几点上班几点下班，它是非常隐性的东西。

我觉得要去做好这样的一件事情，对管理者的要求就是 以身作则。非常明显但是很简单的一个例子，比如开会迟到这个事情在行业里其实还挺普遍的。就是三点要开会，你会发现到了三点零五分，大家还在稀稀拉拉地进会议室。

对此我有一个方法，我每次开会的时候首先我会在群里提示一下大家，然后我会提前两分钟到会议室。我本身提前到了会议室，对大家来讲就是一种震慑，这个其实就是我在给大家立规矩，我没有告诉你说开会要准时，但是大家看到我去会议室了，那就都会跟上。以身作则去完善规章制度指的就是这样的一个过程。

## 给女性程序员的一些成长建议

### 自信起来

感受最深的一点建议还是刚才提到的一个很核心的问题：信心，这是我作为一个女性从业者，从业以来最大的感受。希望女性程序员要自信起

“持续学习，不断进步，这不管是对女性工作者还是男性工作者，其实都是重要的。”

来，勇敢起来。

### 培养抗压性

还有很核心一点就是培养自己的抗压性。这个要心理和身体两个方面都要有，就是你要足够强壮，不管是心理还是体能上。

这个行业对于女性的一些不公平的一些看法、待遇之类的，这些其实都是非常盲目的。你要让自己坦然地接受，然后化解就好。然后如果你整个的心理的抗压跟不上，其实就还蛮麻烦的，至少职业一步步往上走，就需要有强的心理素质，

而身体方面这个就不用说了。

### 持续学习

我工作以来遇到过的大大小小的职业瓶颈，都是通过学习的渡过的。比如我从高级工程师往资深工程师发展的时候，我也遇到了一些瓶颈，但是，通过不断学习，我成功了。现在回头来看，发现其实自己做得还不错。持续学习，不断进步，这不管是对女性工作者还是男性工作者，其实都是重要的。

**张灿**，顺丰科技研发副总监。曾任百度外卖总监。2014 年加入百度外卖，正式开启技术转管理的涅槃之路，在转型专业技术团队管理的职业阶段中巾帼不让须眉，高效完成团队 0-1 搭建、平稳帮助团队快速扩张、梯队搭建等诸多管理工作。在业务推动、技术重点、沟通协作、人才培养、任务分解等诸多方面都快速建立了自己的管理框架，并积累了诸多技术团队管理实战经验，逐步成为了一个素质全面，能力综合的技术团队管理者。

# 赵成：回顾运维的发展历史，我相信AIOps是必然趋势

作者 郭蕾



毫无疑问，运维技术的发展已经进入了深水期。随着 Docker、OpenStack、Puppet 等技术的流行，以及 CI/CD、DevOps 等理念的落地生根，自动化运维的发展迎来了小高潮。整体来看，自动化运维平台帮助提升了运维的效率，并减少了因人工和流程操作而引起的运维故障。

记得在 2001 年的时候，Gartner Group 有一个调查显示在 IT 项目经常出现的问题中，源自技术或产品（包括硬件、软件、网络、电力失常及天灾等）的问题只占 20%，但流程失误方面却占 40%，人员疏失方面也占到了 40%。这些年来，企业通过自动化运维平台以及 DevOps 等协作理念逐步解决了 Gartner 提到的流程失误和人员疏忽相关的 80% 的问题，虽然目前没有具体的统计数据，但可以确认的是，这一问题得到了有效解决。

但另外一个值得注意的点是当前的 IT 项目基础设施环境与 5 年前已经没有办法同日而语，更不用说 10 年前。近几年，随着云计算、微服务等技术的流行，以及互联网业务的迅速发展，运维人员要关注的服务数量也呈现了指数级增长，自动化运维虽然提升了效率，解决了一部分问题，但也遇到了新的难题，比如面对繁多的报警信息，运维人员应该如何处理？故障发生时，又如何能够迅速定位问题？

当企业遇到这些新的问题却无从下手时，恰好历史进入了人工智能时代，那上面的这些问题可否通过『AI + Ops』的这种跨界创新的方式来解决呢？于是 Gartner 在 2016 年时便提出了 AIOps 的概念，并预测到 2020 年，AIOps 的采用率将会达到 50%。简单来说，AIOps 就是希望基于已有的运维数据（日志、监控信息、应用信息等）

并通过机器学习的方式来进一步解决自动化运维没办法解决的问题。

就目前来看，国内的百度、搜狗、宜信、阿里巴巴都已经探索尝试了AIOps，并且取得了不错的收益。在2017年InfoQ举办的CNUTCon全球运维技术大会上，也有不少AIOps相关的议题，甚至会议主题也从去年的容器生态迭代到今年的智能时代的新运维，感兴趣的读者可以关注。

那对于AIOps这个新名词，它又会涉及到哪些新技术？从运维的发展角度看，为什么说AIOps是必然趋势？它与自动化运维之间会有什么样的关系？InfoQ记者就这些问题采访了美丽联合集团运维经理赵成。

#### InfoQ：如何理解AIOps？AIOps会涉及哪些技术？这又是一个新名词吗？

**赵成：**我觉得理解AIOps之前，还是先理解下AI、机器学习、深度学习这样几个概念。如果用一张图，来表示，就是图1。

同时，这张图的来源文章，也建议看一下，

讲的还是比较清晰的。

简单来说，AI，人工智能是一个广义概念，最早期提出来的时候，人们的愿景是希望AI能够完全具备人类智慧，这属于“强人工智能（General AI）”。但是除了在科幻电影和科幻小说外，在现实中还没有实现，可能不仅仅是要求对计算机技术要求极高，对生物医学技术也会有很高的要求，因为前提是得弄懂人类大脑运转的每一个细节吧。

但是过程中，研究人员发现我们虽然没法让机器完全具备人类一样的智慧，但是在非常具体和特定的领域，机器是可以做的比人类更好的，比如图形图像处理、语音识别等等，这些人工智能的应用，称之为“弱人工智能（Narrow AI）”，这些应用的实现手段，就得益于机器学习算法长足的进步，而深度学习又是机器学习领域很精深的一部分。

所以，我们现在提到的AI，更多的是依赖机器学习（包含深度学习）算法的实现的AI场景，

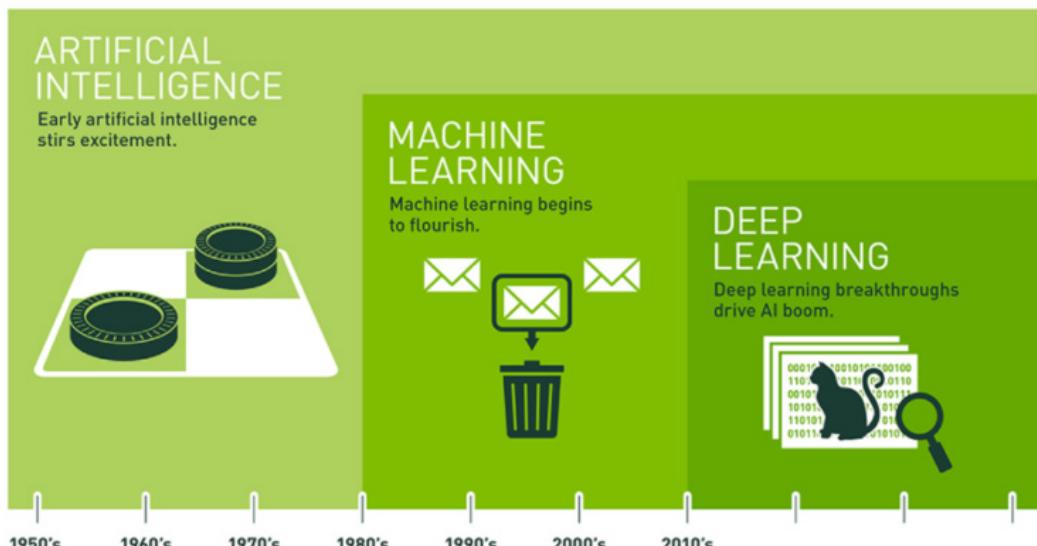


图1

或者说机器学习算法只是实现 AI 的其中一种手段。

了解了上面的概念，再回到 AIOps 上来，拆分为 AI + Ops 会准确一些，也就是 Ops 与 AI 相结合可以做的事情。Gartner 的定义是 Algorithmic IT，而不是 Artificial Intelligence，我起初觉得也不是很合理，但是我认真理清楚整个关系之后，我觉得这样定义也没有问题。

最后，我觉得定义如何到没有必要纠结，因为不管 AIOps 里这个 AI 到底是 Algorithmic IT 还是 Artificial Intelligence，最终，我们根本上使用的，还是机器学习算法这个手段。

AIOps 涉及的技术，从 AI 的角度，主要还是机器学习算法，以及大数据相关的技术，因为涉及到大量数据的训练和计算，从 Ops 的角度，主要还是运维自动化相关的技术。另外 AIOps 一定是建立在高度完善的运维自动化基础之上的，只有 AI 没有 Ops，是谈不上 AIOps。

**InfoQ：你认为 AIOps 是运维发展的必然趋势吗？从手工运维，到自动化运维，再到现在的 AIOps，谈谈你理解的运维发展趋势？**

**赵成：**必然趋势。一个很明显的规律，凡是能让我们的生活变得更美好、更简单、更方便的技术，一定会具有强大的生命力，也必然会成为发展趋势，而 AI 正是这样的技术之一，AIOps 又是其中的一个专业领域。

运维的发展变化，我的感受，是随着业务和技术发展变化的，根本上还是业务驱动和倒逼出来的。

比如 2008 年 -2010 年，我接触的是电信级软件的开发和维护，那个时期的软件有这么几个特点：

- 业务场景和形态上，相对固定，变化不大；
- 软件是分层架构，模块数量固定，架构上基本不会有太大的变化；
- 研发流程和规范非常严格，基本交付上线之后不会有太大的功能和性能问题，线上出问题会面临非常严格的处罚；
- 软件交付周期长，大的变更半年一次，小变更至少 3 个月，这期间研发会将发布脚本以及产品升级文档制定的非常详细，我们称之为 Step by Step，而且也是经过严格验证和测试的；
- 研发维优团队支持，一旦线上出现软件方面的问题，一线维护直接将问题转回研发进行定位处理，而且有严格的 SLA 约束。

这种情况下，变化不大、研发可以搞定绝大部分软件层面的事情，所以我们就会看到这个时期的运维更多的是网络、硬件、系统方面的维护职责，这个是由那个时期的业务特点，以及软件架构特点所决定的。

当然这并不代表那个时候业务的技术难度和复杂度不高，反而技术门槛是相当高的，协议之复杂，网元交互之多，架构图和交互逻辑画出来也是相当复杂，那个时候我们答辩和述职必做的事情就是看网讲网，要对所有的周边网元以及交互关系讲清楚。

之后，随着互联网业务的高速发展，业务场景上丰富多彩，复杂和多变，新业务和新场景也在不断涌现，为了快速验证产品和需求的方向是否有效，就要快速试错，对迭代开发和交付效率有了极高的要求，所以技术界逐渐催生出了服务化这样的软件架构，以及持续交付过程，同时随着业务体量快速膨胀，衍生出对稳定性有极高的要求，这时我们现在常听到的全链路跟踪、容量

“ AIOps 的首要前提条件，一定是先要有高度完善的运维自动化，自动化都没做好前，先不要玩 AI，千万不要本末倒置。 ”

评估、限流降级、强弱依赖等稳定性的解决方案就涌现出来。

这时对于开发的诉求是能够将更多的精力放到需求实现上，而因服务化带来的大量的应用管理、持续交付、监控、稳定性、成本控制等非功能性体系的建设和保障就需要有专门的团队来做，这时对于运维的诉求也在悄然发生着变化，所以这也是为什么这个阶段会涌现出 DevOps、技术运营、PE、SRE 等等对运维重新定位的词汇。当然这个过程中，因为云计算发展，传统的网络、硬件和系统维护的职责在逐渐的被弱化，也在逼迫着运维的关注点从底层转向应用和业务层面。所以，我们看到就在近 2-3 年，自动化、发布系统、稳定性平台这些系统成为了运维团队重点关注和建设的部分。

这里就有一个趋势，就是 SpringCloud 和 Cloud Native，SpringCloud 让当前服务化的开发变得越来越方便和高效，而 Cloud Native 在

打造应用的基础设施方面也已经取得了长足的进步，而且已经有了 CNCF 这样的组织在驱动响应的标准和体系建设。设想一下，后面从业界的角度，如果 Spring Cloud 成为微服务的开发标准，Cloud Native 成为应用的运维标准，是不是又会驱动着一波运维的转型和升级呢？

回到 AIOps 上来，当前这个阶段，现实情况，系统里面已经有大量软硬件模块、日志、监控告警指标也纷繁复杂，一方面是无法在问题萌芽状态就发现问题，无法提前做出预判，另一方面是发生了问题又无法快速确定根因，造成持续的资源损耗。技术发展上，随着计算能力、数据量的积累、以及机器算法的进步，如何更加高效的开展 Ops 这个问题就摆在我面前，AIOps 的模式应运而生。

所以，运维一步步发展到当前这个状态，根本上讲还是业务高速发展倒逼出来的，同时，从手动运维到运维自动化，再到 AIOps，这个过程

根本上是在朝着如何更加高效运维的趋势在发展。

### InfoQ：AIOps 的出现是为了解决哪些问题？这些问题运维自动化无法解决吗？

**赵成：**主要还是解决复杂环境下问题的快速发现甚至提前预判，以及出现问题后的如何在复杂的告警、报错和日志中快速进行根因分析。

运维自动化无法解决吗？我的理解，AI 和 Ops 要解决的还是两个层面的问题，可以类比到人，AI 相当于人的大脑，我们手脚和躯干是执行系统，大脑负责决策判断，手脚躯干负责完成大脑下发的动作指令。对应到运维上面，AI 要解决的是怎么快速发现问题和判断根因，而问题一旦找到，就需要靠我们高度完善的自动化体系去执行对应的运维操作，比如容量不够就扩容、流量过大就应该触发限流和降级，等等。

AI 是能够让 Ops 执行的更加高效的强大助推力，下面是我之前整理出来的，我理解的 AIOps 的体系和建设思路（见图 2）。

### InfoQ：落地 AIOps 的前提条件是什么？什么样的团队适合落地 AIOps？

**赵成：**AIOps 的首要前提条件，一定是先要有高度完善的运维自动化，自动化都没做好前，先不要玩 AI，千万不要本末倒置。

从 AI 的角度，应该有三个方面的充要条件：机器学习算法、计算能力（类似于 GPU 的高性能设备）、海量数据。

先看算法，这个基础，在 AI 中我们使用到的各类算法，比如基于指数平滑的二次平滑、三次平滑算法，基于分解的傅里叶分解、小波分解算法等，基于深度学习的前馈神经网络、循环神经网络 RNN 算法等，这些算法早就已比较成熟了，并大量使用在其它的研究领域，比如我们熟悉的

图 1 第一阶段：发现问题



图 2 第二阶段：分析问题

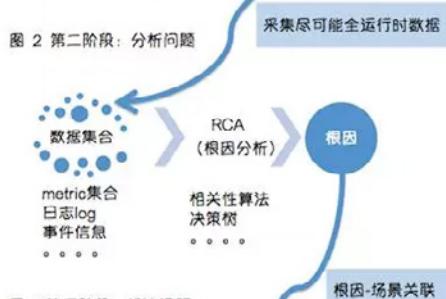


图 3 第三阶段：解决问题

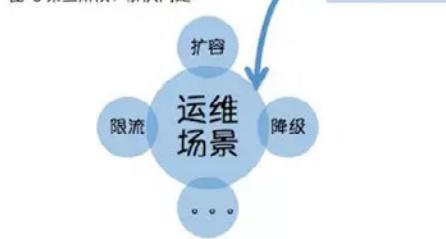


图 2

图形图像处理、语音识别，还有在医疗、电力以及通信行业的应用。所以在算法上，我们很早之前应该具备了这方面的理论基础。

但是 AI 为什么这几年突然火起来，或者在应用上有了长足的进步，很大原因就是计算能力提升了，海量数据积累起来了。比如随着硬件计算能力的提升，有了 GPU 这样的超高性能计算设备，同时还有云计算这样规模的基础设施支持；再就是，最近这些年随着互联网业务的高速发展，各行各业都积累了海量的现实的数据。

比如吴恩达教授非常著名的识别猫的实验，一方面吴教授有自己非常牛的深度学习算法，另外一方面，基于 16000 个处理器的计算能力搭建起了深度学习的平台，然后基于 Youtube 中千万级别的猫的视频图像进行算法的学习和训练，最终可以非常精确的从图像中识别出猫。

回到 AIOps 上面来，看这个三个条件：

- 算法还是那些算法，不过得要有相应专业能力的团队，如果是纯应用，我觉得运维团队倒是可以自学一下，但是不管怎么样，这个还是有一定门槛，需要大量的学习和能力提升。
- 数据就是要靠线上运行的真实数据和日志，所以必须要有大量的数据积累
- 计算能力上，目前看到我们基于大数据技术的数据处理能力已经足够，因为毕竟不是像图形图像这样的复杂计算场景。

从上面三个条件看，也就不难理解，AIOps 做的比较超前的为什么都是那些国内外的大厂，因为有技术实力、有足够的资源、有足够的数据，最关键的是足够复杂和变态的业务场景以及运维场景，在倒逼着 Ops 往这个方向上走。

至于什么样的团队适合落地，我暂时没想到什么标准，不过还是那个建议是，先尽快做好自动化，把基础打好，AI 的学习上做一些储备，当业务复杂度和体量到达一个量后，会自然倒逼着运维往这个方向发展，千万别自动化还没做完完善，就跟风搞 AIOps。

#### **InfoQ：AIOps 中的数据是怎么来的？数据是必要的吗？**

**赵成：**AIOps 中的数据必须是线上产生的现实场景下的运行数据，不管是底层硬件和系统层

面，还是应用和业务层面，以及运维的操作记录日志，要尽可能全面的数据。这些数据一方面要做算法模型的训练，让算法能准确识别问题，一方面还要在问题分析时做根因分析使用。

数据是必要的，准确说是必需的。目前 AIOps 中，就异常发现来说，针对不同的应用场景，应该使用哪种算法模型，这个还是有一定挑战的，所以起初可能会同时使用多种算法同时运行，这时就需要大量真实的数据去验证算法运行的情况，同时做一些参数校正，也就是我们所说的训练的过程，最终我们根据跑出来的结果准确度选择合适的算法，或者设定相应的权重。所以，机器学习算法是否有效是离不开大量的真实数据的训练的。

#### **InfoQ：可否谈谈你们的 AIOps 落地场景？**

**赵成：**这块我们还在实践中，一块是异常检测，做一些关键监控 Metrics 的曲线监控，这块用到的基本是常见的指数平滑、3-Sigma 算法等。另一块是根因分析，在服务化的架构中，最头痛的还是出现了故障，无法快速的定位原因。大致思路是，根据全链路跟踪系统的每一次请求的依赖关系，做调用的关联度分析，当一个模块出现问题时，会同时导致依赖这个模块的所有模块都会告警，甚至还有业务层面的告警，这时就需要快速的根因分析，确定问题在哪儿。

**赵成（谦益）**，美丽联合集团运维经理，负责美丽联合集团（原蘑菇街、美丽说）运维团队管理及运维体系建设工作。拥有近 10 年研发和运维经验，见证和参与了多个电信级和互联网产品从无到有的建设，从微量到海量的成长过程，拥有非常丰富的电信级和互联网业务研发和运维经验。目前专注于运维创造价值，以及云计算和 AI 时代运维的转型和突破。个人公众号 Forrest 随想【微信号：forrest\_thinking】，欢迎关注。

# 腾讯游戏容器云平台的技术演进之路

作者 郭蕾



从 2014 年开始，腾讯游戏就逐步在生产环境中使用容器相关技术，到现在，代号为 TenC 的容器平台支撑了近 200 款游戏的运营。在这 3 年的时间里，整个平台经历了从最开始的“轻量级虚拟机”使用 Docker 的方式，到现在的原生容器云方式的迭代，接入的业务也由原来的在线服务扩展到现在的微服务、大数据、机器学习等类型。

2015 年时，InfoQ 记者曾对腾讯游戏进行过一次专访，在提到容器对于游戏业务的价值时，腾讯游戏高级工程师尹烨这样说道：“相比其他行业，游戏业务更为复杂多样，有端游、手游、页游之分，同时，还要分区和分服，甚至还要区分自研和代理。这些特性给运维和部署带来了诸多不便，而 Docker 统一的镜像分发方式，可以标准化程序的分发部署，提高交付效率。另外，

游戏业务的生命周期长短不一，这也就需要弹性的资源管理和交付，而容器更加轻量，资源的交付和销毁更快。”

而在当时，Kubernetes 刚刚发布 1.0 版本，尹烨也表示他们对于 Kubernetes 的应用还没有完全发挥其优势，接下来一段时间，他们也会探索容器平台与业务开发、运维的结合方式。那在时过境迁的两年之后，腾讯游戏在容器平台的实践方面又有哪些经验和心得？InfoQ 记者再一次采访了尹烨。

**InfoQ：2014 年的时候腾讯游戏就开始在生产环境中使用了 Docker，到现在已有 3 年时间。能否谈谈你们从最初的简单使用 Docker 到现在的容器云平台，这中间你们大致经历了哪几个阶段？**

**尹烨：**现在我们的 Docker 容器平台（内部



代号：TenC）主要支撑了如下三种类型的业务场景，也基本上对应了平台发展的三个阶段，每个阶段都是紧随着业务的需求在推进。

**第一阶段：**我们把容器当作轻量虚拟机来给业务使用，让业务程序不需要特殊修改，就可以在容器中正常运行起来，我们花了很多精力来兼容原有业务架构和使用习惯，以及保证底层运行环境的稳定运行；

**第二阶段：**后来在代理游戏中，出现一些微服务架构的业务，这些业务完全采用原生的 Docker 方式，基于 Docker 镜像来做分发和部署，弹性伸缩；

**第三阶段：**再后来随着大数据和 AI 兴起，这类业务计算量非常大，资源需求要求高弹性，使用 Docker 容器来交付资源相对于原来物理机、虚拟机方式效率高很多。

**InfoQ：目前腾讯游戏有多少的业务跑在容器云上？可否谈下你们的容器云技术栈？**

**尹烨：**目前有过百款游戏业务都运行在容器上。在游戏业务接入容器平台前，会有一些评估要素。评估通过的话，新业务基本上会首选放到容器里。我们的平台主要基于 Docker、Kubernetes（后面简称 K8s）等开源组件开发的。当前 Docker 主要使用了 1.12 的版本，Kubernetes 主要使用 1.2/ 1.5 版本。在实际的应用过程中，我们会基于我们自身的基础环境做一些二次定制开发，包括资源调度和网络部分，例如我们自己开发的 SR-IOV CNI 插件等。整体结构大致如图 1 所示。

其中 TLinux 是腾讯操作系统团队自研的服务器操作系统，操作系统团队针对 Docker 平台需求，进行了一系列内核特性研发和针对性改进。

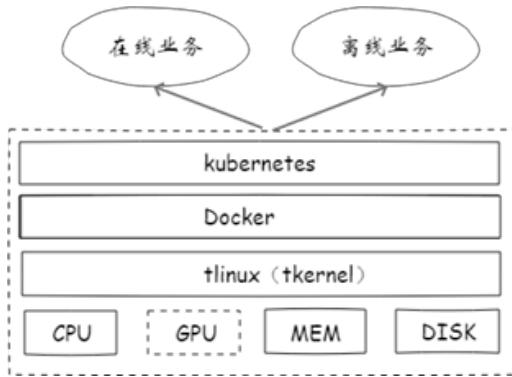


图 1

TLinux 为游戏在 Docker 容器上的稳定运行奠定了基础。例如，

cgroup 高版本特性移植，如 cgroup namespace, pids cgroup, cgroup 支持 I/O 隔离等；

aufs、overlayfs 特性移植；网络 sysctl 隔离；

其他诸如 cgroup、overlayfs、xfs 等等的 bug 修复。

**InfoQ：记得 2015 年的时候你们是基于 Kubernetes 0.4 版本做的改造，现在 Kubernetes 也跟着升级了？升级之后之前哪些改动是怎么处理的？现在还有基于主干版本做定制吗？**

**尹烽：**2014 年我们开始使用 Docker 时，Google 刚刚开源了 K8s，K8s 的设计非常精巧，我们就开始做了一些定制，来管理容器。我们现在还在使用那个古老的深度定制版本来管理我们的轻量级虚拟机容器，因为这类业务对编排没什么需求，为了保证平台的稳定，一直没有升级，将来也不太会。但针对微服务和离线计算这部分业务，我们没有太多侵入式的定制修改 K8s，主要通过插件方式来扩展 K8s，我们会根据自身的需求来跟随社区升级 K8s。

在功能定制方面，先大概介绍几点吧。比如，我们结合业务需求，和自身的基础环境，主要在调度和网络等方面进行了一些扩展。首先，在网络方面，我们开发了 SRIOV CNI 插件。

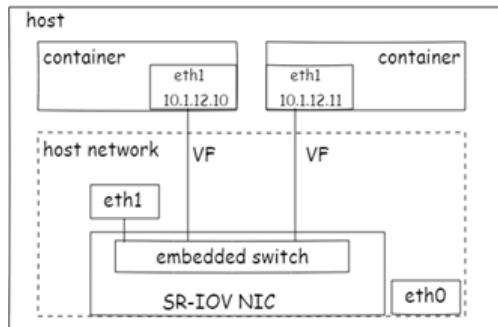


图 2

1. 母机上开启SRIOV功能，同时向网管系统申请子机IP资源，每个VF对应一个子机IP。
2. Kubernetes在调度时，为每个Pod分配一个VF与子机IP。
3. 在Pod拿到VF与IP资源，进行绑定设置后，就可以像物理网卡一样使用。

同时我们也做了一些优化：比如将 VF 中断绑定到容器分配的 CPU，并开启 RPS，把网卡的软中断分到各个 CPU 处理，来提升网络性能。

其次，在调度方面，为了支持 SRIOV 插件，在容器调度中，除了 K8S 原生提供的 CPU、Memory、GPU 之外，我们还把网络（物理 IP）也作为一种资源来调度，同时结合 K8S 提供的 extender scheduler 接口，我们定制了符合我们需求的调度程序（cr-arbitrator）。其结构如图 3 所示。

cr-arbitrator 做为 extender scheduler，集成到 K8S 中，包括两部分内容。

- 预选算法

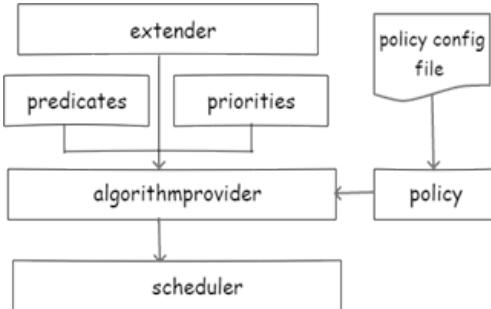


图 3

在完成 Kuernetes 的 predicates 调度后，会进入到 cr-arbitrator 的预选调度算法，我们以网络资源为例，会根据创建的容器是否需要物理 IP，从而计算符合条件的 node（母机）。

- 优选算法

在整个集群中，需要物理 IP 的容器与 Overlay 网络的容器并未严格的划分，而是采用混合部署方式，所以在调度 Overlay 网络的容器时，需要优化分配到没有开启 SRIOV 的 node 上，只有在资源紧张的情况下，才会分配到开启 SRIOV 的 node 上。

除了 cr-arbitrator 实现的调度策略外，我们还实现了 CPU 核绑定。容器在其生命周期内使用固定的 CPU 核，一方面是避免不同业务 CPU 抢占问题；另一方面在稳定性、性能上（结合 NUMA）得到保障及提升，同时在游戏业务资源核算方面会更加的清晰。

**InfoQ：内部有游戏开始使用微服务架构了？你们的容器云平台是如何支撑微服务架构的？**

**尹烽：**腾讯游戏主要分两部分，一部分是代理的，另外一部分是自研的。对于自研的业务，经历那么多年的沉淀，有一套非常成熟的框架和运营体系。微服务架构带来的变化比较大，已有

的业务架构转过来成本太高。

而代理游戏进行架构的转变，相比而言成本会低很多。所以，一些新的代理业务往往会采用微服务架构。今年开始，我们内部也有一些在线业务也在开始这方面的尝试，我想后面这种业务会越来越多的。

这些业务本身的架构使得业务很容易在容器上运行。另外，如果业务需要使用 K8s 的一些高级特性，比如服务发现、自动扩缩容机制等，则需要业务在架构上和实际部署时做一些简单的适配，这些适配都比较简单。

相对于轻量虚拟机，微服务的业务在网络、监控、日志等方面有较大的不同。

首先是网络部分，受限于底层物理网络，业务的容器不再每个容器一个物理 IP（这会消耗大量的内网 IP 资源，而且管理也不方便），所以，我们使用了 Overlay 的网络方案，将容器的网络与底层物理网络解耦。业务的逻辑部分的容器都会跑在 Overlay 网络上，但是业务的数据层服务，比如 MySQL, Redis 之类的，仍然跑在物理网络上，Overlay 与数据层之间通过 NAT 访问。同时，接入层通过内部 LB 对接外部网关，分离网关与业务服务。大致结构如图 4。

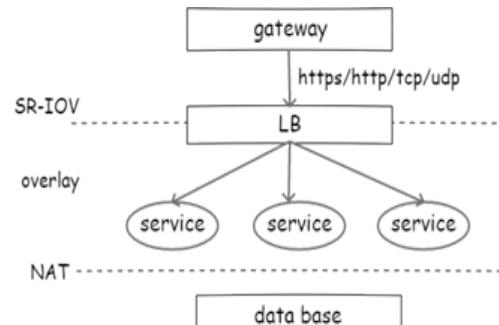


图 4

其次是监控与告警，对于微服务业务，无法使用原来的监控方案。而监控、告警是整个游戏运营过程中最为核心的功能之一，在游戏运行过程中，对其性能进行收集、统计与分析，来发现游戏模块是否存在问題，负载是否过高，是否需要扩缩容之类等等。在监控这一块，我们在cAdvisor基础上进行定制，其结构如下图5所示。

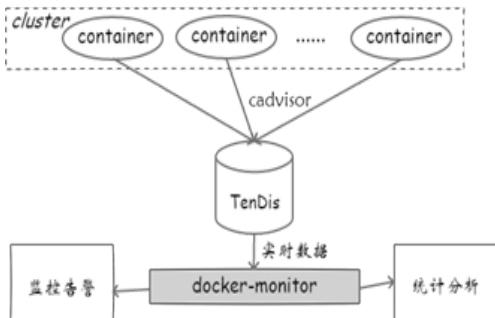


图 5

- 每个母机部署cAdvisor程序，用于收集母机上容器的性能数据，比如CPU使用情况、memory、网络流量、TCP连接数等。
- 在存储方面，目前直接写入到TenDis中，后续如果压力太大，还可以考虑在TenDis前加一层消息队列，例如Kafka集群。
- Docker-monitor，是基于cAdvisor收集的数据而实现的一套性能统计与告警程序。在性能统计方面，除了对每个容器的性能计算外，还可以对游戏的每个服务进行综合统计分析，一方面用于前端用户展示，另一方面可以以此来对服务进行智能扩缩容。告警方面，用户可以按业务需求，配置个性化的告警规则，docker-monitor会针对不同的告警规则进行告警。

最后，是业务日志的处理。Docker在容器

日志处理这一块，目前已很丰富，除了默认的json-file之外，还提供了gcplogs、awslogs、fluentd等log driver。而在我们的日志系统中，还是简单的使用json-file，一方面容器日志并非整个方案中的关键节点，不想因为日志上的问题而影响Docker的正常服务。

另一方面，把容器日志落地到母机上，接下来只需要把日志及时采集走即可，而采集这块方案可以根据情况灵活选择，可扩展性强。我们当前选择的方案是Filebeat + Kafka + Logstash + Elasticsearch，其结构如图6所示。

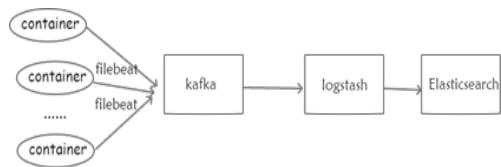


图 6

我们以DaemonSet方式部署Filebeat到集群中，收集容器的日志，并上报到Kafka，最后存储到Elasticsearch集群，整个过程还是比较简单。而这里有个关键点，在业务混合部署的集群中，通过Filebeat收集日志时怎样去区分不同的业务？而这恰恰是做日志权限管理的前提条件，我们只希望用户只能查看自己业务的日志。以下是具体的处理方案与流程：

- 首先我们在Docker日志中，除了记录业务程序的日志外，还会记录容器的name与namespace信息。
- 接着我们在Filebeat的Kafka输出配置中，把namespace作为topic进行上报，最终对应到Elasticsearch的index。
- 在我们的平台中，一个namespace只属于一个业务，通过namespace，可以快速的搜索

到业务对应的日志，通过容器的name，可以查看业务内每个模块的日志。

**InfoQ：这么长时间的应用，有做过复盘吗？**  
**未来有什么计划？**

**尹烨：**我们一直致力于为游戏业务提供高效的计算资源。从轻量虚拟机，到微服务，再到离线计算，我们一直紧随业务的场景往前演进。特别是弹性计算场景，相对于原来交付物理机 / 虚拟机的方式，现在基于容器的方式更加高效。我们希望在满足业务对计算资源需求的同时，尽可

能提高资源的利用率，从而降低运营成本。我们会一直朝这个目标努力。

具体来说，整个平台也还有很多需要改进和优化的地方。比如，网络方面，我们希望简化外部网关的接入，进一步提升底层 overlay 网络的性能。优化资源调度策略，考虑更多的一些因素，比如任务优先级、任务运行时长等，将在线业务与离线业务混合部署等。

**尹烨**，2011 年毕业后加入腾讯，先后从事后台服务、MySQL 内核开发；14 年开始从事 Docker 等容器技术研究，并负责 Docker 等容器技术在腾讯游戏业务的实践和落地。个人喜欢研究各种开源技术，对 Linux 内核、Docker/Kubernetes 等有较为深入的研究。

# 深度学习打造精准推荐系统，细说国美互联网AI发展的进击之路

作者 蔡芳芳



这是一个 AI+ 的时代。作为线上 + 线下的电商零售平台，国美互联网如何将人工智能技术嵌入到实际业务中？机器学习和深度学习技术为国美带来了哪些改变？在这火热的时局中，国美未来在前沿技术方面又将如何布局？且听下文一一分解。

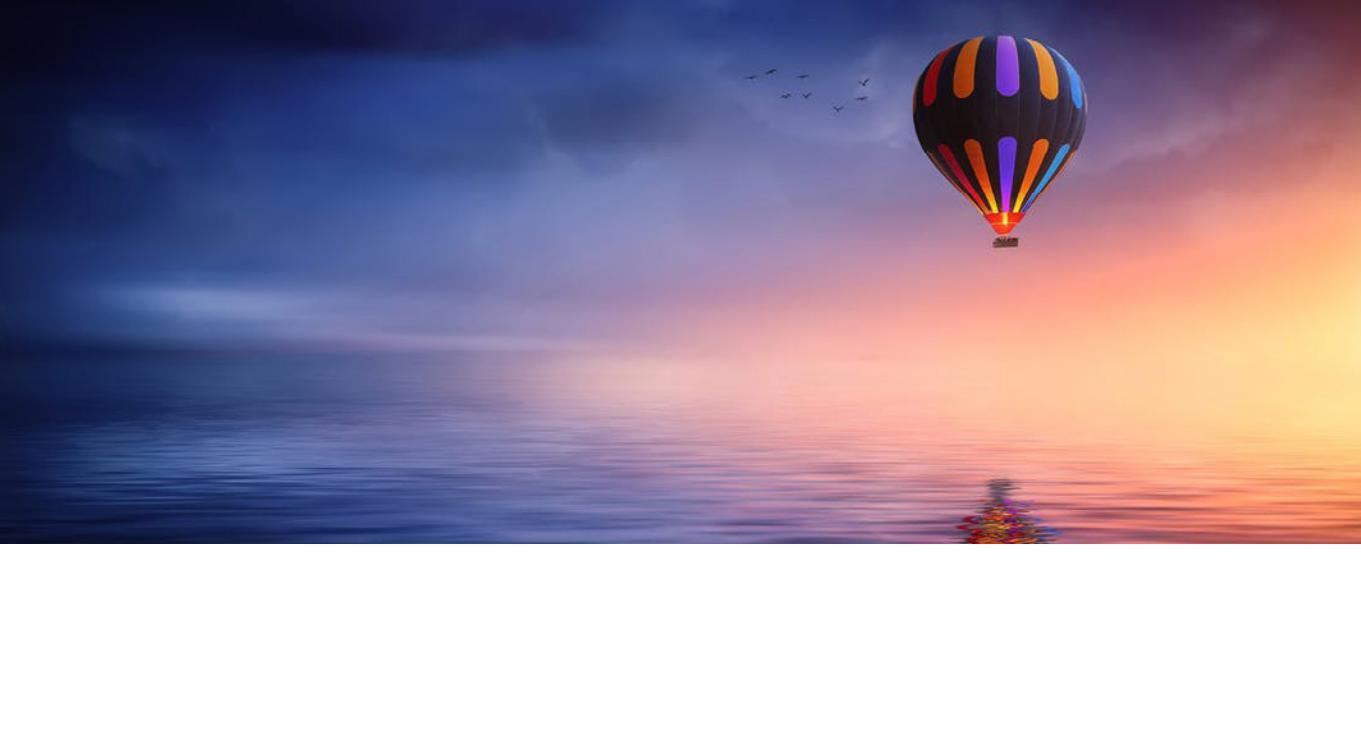
今年机器学习已然成为炙手可热的技术话题。深度学习与人工智能技术正在改变人们的生活，同时也给企业管理大量数据、为用户提供更精准的服务提供了一些新的思路和尝试的方向。越来越多的企业开始尝试将机器学习引入原有的大数据平台和框架中，并将公司业务与机器学习、深度学习技术相结合以寻求更优的业务价值。

2016 年 11 月国美互联网板块所属国美在线、美信、国美管家、GOME 酒窖、国美海外购整合组建国美互联网生态（分享）科技公司（下面简称国美互联网），并推出了国美 Plus APP。作为具备线上线下双重基因的新零售平台，

国美互联网近年来，针对机器学习、深度学习技术与社交电商业务相结合进行了一系列尝试和创新。这次我们非常有幸对国美互联网大数据中心的副总监杨骥进行了采访，目前他主要负责搜索、推荐和深度学习三个方向的研发工作及团队管理。作为国美互联网个性化推荐团队和算法架构平台建设者，杨骥与我们分享了国美互联网在机器学习技术之路上的尝试、挑战与收获。

## 国美大数据中心背景

国美互联网大数据中心主要负责大数据平台运维、大数据应用、BI、搜索、推荐等相关内容，



支撑国美互联网全业务线（包括大家电、百货、汽车等）的精准营销和数据变现。尤其是最近几年以来，中心在机器学习方面投入了非常多的资源，除了升级推荐和搜索相关的机器学习算法之外，还陆续推出了“国美拍照购”、“图像相似推荐”、“图像搭配推荐”等基于深度学习算法的产品。

## 国美个性化推荐系统面面观

对于电商来说，个性化推荐系统是不可或缺的利器，也是国美互联网应用机器学习和深度学习技术的主要战场。

### 推荐系统演进之路

要将机器学习技术应用到实际产品中，必然离不开一套良好的算法和平台架构。2016年3月杨骥加入国美之后，对机器学习和深度学习算法以及

线上服务架构进行重构，打造了全新的个性化推荐系统。目前，杨骥带领团队已经完成了三轮大规模的架构升级，最终实现了一套完整的推荐系统架构，其中包括机器学习和深度学习计算平台，并整合了A/B测试、线上部署和自动化监测等功能。

### 系统架构和流程

图1和图2是国美互联网应用 Learning to Rank（机器学习排序，L2R）后的推荐系统流程图和架构图。采用的工具包括 Apache Storm、Kafka、Spark、Flink 等。

当用户在网站或者 APP 上看到“猜你喜欢”的商品后进行了点击。一方面，该点击信息通过实时数据处理系统反馈给排序引擎和在线训练模型，帮助推荐系统实时调优；另一方面，点击信息进入 Log。基于 Log 进行知识挖掘，获取相关特征和训练数据，并进行基础推荐模型的计算

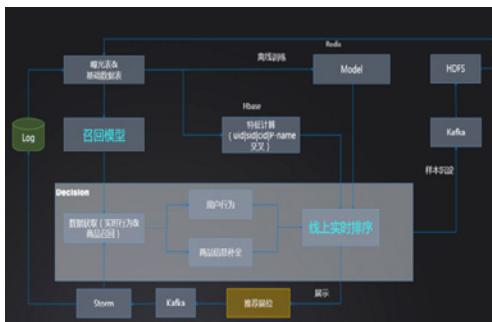


图 1

（如关联规则和协同过滤模型等）以及离线排序模型的训练。离线排序模型缓存到 Redis 模型服务器，而相关特征数据、基础推荐模型的粗筛结果缓存到 Hbase 和 Redis 数据服务器。

系统要向用户展示推荐商品时，数据获取模块获得用户的基本特征和推荐的初筛结果，进行交互特征和在线特征的计算，然后利用离线模型、在线模型以及业务规则进行最终的筛选与排序，返回最终推荐结果给用户。

## 数据收集和清理

国美使用 Apache Kafka 和 Apache Storm 系统收集和处理推荐展位上的实时数据消息，包括推荐展位每页已经展示过什么商品、用户点击了什么商品等。然后对这些数据消息进行实时分析和统计，包括用户在推荐展位展示之前已经对哪些品类的产品有过购买行为，对不同的已购买商品赋予不同的权重（比如手机买过则短期不会再买，纸巾则会隔较短时间反复购买），再计算出接下来的推荐展位应该显示哪些商品。

对于一些特殊的用户数据需要进行数据清理，以免影响算法模型的拟合效果。比如，已购买用户对同一商品的点击和查看对推荐排序模型是无用的，应该被排除；还有一些用户频繁地点击商品但却不下单，被称为“点击狂人”，可以

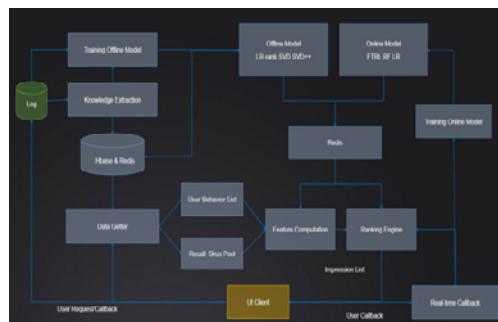


图 2

通过统计方法将其剔除。

## 离线模型 + 在线学习

国美互联网的推荐系统中离线模型和在线学习相辅相成。

离线模型主要是为了节省计算时间、加快响应速度，当然需要消耗一些缓存空间并牺牲一定的精度。例如，召回模型中会预置一些算法模型，并根据快照信息离线进行商品的初筛，使进入实时排序模型的商品量从千万级别减少为数百个，大大提高了实时排序的效率。目前国美互联网的召回模型中有 20-30 个算法模型，既保证了召回的多样性，也不至于给维护带来太多麻烦。

而在在线学习则是为了提高排序的准确度，它能根据用户实时的行为数据对模型进行实时训练，使当前模型准确地反应用户当下的兴趣和倾向。但是使用在线学习也会引入新问题，比如在电商店庆日，用户的购买行为可能是非理性的，如果一味地使用在线学习将会给算法模型引入严重的偏差，因此国美的推荐系统会定期（每天或每隔几个小时）使用离线数据进行模型训练并对在线模型进行校准。

## 用户画像

对于电商来说，内部收集到的用户数据用来

做用户画像肯定不如社交类网站的数据丰富，可能导致做出来的用户画像不够立体。因此国美互联网通过将用户行为映射到商品信息上来构建用户画像。比如用户行为（点击、关注或收藏等）涉及的一系列商品的属性（比如品类、品牌、中心词、价格等）作为用户行为的映射，刻画出用户感兴趣的商品，然后再结合用户标签（比如有车一族、户外运动爱好者等），完成用户画像的构建。目前用户画像在国美互联网主要用于商品召回和最终商品推荐时的过滤。

## 算法评测

对于个性化推荐排序来说，设定可量化的目标是非常重要的。

国美对算法的评估方式包括离线评测、在线 A/B 测试和算法覆盖率测试。离线评测常用的评估指标为 AUC、Logloss 和 NDCG，其中 AUC 和 Logloss 主要评估分类的准确率，而 NDCG 则是评估排序质量的指标。

## 实际效果

国美互联网采用 CTR/CVR/GMV 来对推荐排序算法的实际效果进行评估。与传统的推荐方法相比，使用机器学习和深度学习技术后的个性化推荐排序给国美互联网的 GMV、点击率和转化率都带来了更为显著的提升。

2017 年以来，凭借深度学习技术，离线模型和实时排序模型的质量大幅提升，尤其是 1-5 月份与 2016 年 1-5 月份相比提升效果十分显著，推荐 GMV 提升 70%、参与转化率 CVR 提升 100%。而 CTR 也有一组数据可供参考，采用个性化排序后，PC 订单页“猜你喜欢”展位 CTR 提升幅度 30.79%，PC 首页“猜你喜欢”展位 CTR 提升幅度 14.16%。

大数据 + 机器学习 & 深度学习技术的力量着实惊人。

## 深度学习在国美的应用实践

自 2012 年 ImageNet 大赛技惊四座后，深度学习已经成为近年来机器学习和人工智能领域中关注度最高的技术。

### 国美 Plus 拍照购

国美互联网也展开了深度学习在图像识别中的研究，并且已经应用于国美 Plus 的拍照购功能，它可以根据用户上传的图片预测品类，推荐相关商品（图 3）。

当用户拍照或者从 PC 和手机选取商品图片上传成功以后，系统就会进行特征向量的计算及匹配。然后猜测用户想要的商品，根据商品图片库的相关结果进行展示（图 4）。

图像检索流程可以分为四步，分别是图像标注、提取特征、降维和匹配检索。

- **图像标注：**对国美全站的商品图像进行标注，最主要的工作是数据清洗，即把不能和品类对应的图片删除或者重新进行品类校准；先统计国美全站最近一年内各品类下商品总数的分布，然后按照分布进行图片抽样。
- **提取特征：**图片的特征分为通过深度学习得到的特征和图像局部特征。可利用 caffe 对 CNN 进行训练，将倒数第二层输出作为 Feature Learning 的结果提取出来（Deep Learning + Transfer Learning）；可利用局部特征算子（SIFT、kaze 等）提取出图像的局部特征。
- **降维：**用积量化等方法对深度学习特征和图像局部特征进行编码和降维。



**图 3**

- 匹配检索：采用最近邻搜索的方法找出每一个商品的相似商品集合。

## 图像识别与推荐

基于拍照购，国美还将图像识别融合到了个性化推荐、相似推荐和搭配购推荐中。

### 用户行为 + 商品内容 = 更精确的推荐

杨骥认为，电商领域传统的基于用户行为构建个性化推荐系统存在一定的局限性。

基于用户行为的推荐系统以用户的浏览、点击、收藏、关注、加入购物车、下单等行为作为数据基础，来分析用户可能的购物倾向。对于用户来说，从进入网站或 APP 开始浏览商品、到考虑是否购买、最终到下单是一个连续的过程，但对于算法工程师来说得到的只是日志中几个离散的记录，信息量严重不足，通过这些有限的信息难以推测用户完整的购物决策过程。杨骥认为，除了用户行为，还应该重点考虑商品的内容（比如商品信息主图、商品详情页图片、标题、广告用语和风格等）能够对用户产生多大的冲击。目前国美互联网推荐团队的工作更多地侧重于内容方面。

国美互联网搜索推荐团队利用深度学习技术（如 CNN 卷积神经网络、目标检测等）对商品



**图 4**

主图、详情页图片进行分析和描述，将图片（非结构化数据）转成文字（结构化数据）并存放到数据库里，以便后续分析时调用。他们由此构建了一个基于图像识别的相似召回模型，将其与基于用户行为的召回模型融合在一起，在此基础上进行线上交互。如此一来得到了更多刻画用户、刻画商品的维度，提升了召回模型的精准度和多样性，效果优于仅仅关注用户的点击、关注、购买等行为。

“推荐系统并非 0 和 1 的问题，而是要最大化用户购买概率。先满足 80% 用户的大体需求，然后再对 20% 的用户进行精细的需求雕刻。”

## 迁移学习辅助小数据集训练

利用机器学习和深度学习构建推荐场景时，最大的难点是数据源头，而深度学习对训练数据量要求很高，并不是每个机器学习问题都有足够多的训练数据。因此国美也尝试将深度学习与迁移学习（Transfer Learning）结合起来。

仍以拍照购功能为例，拍照购涉及到图像相似性检索的问题。而已经有很多人针对已有的海量数据训练出了不同任务场景下的模型，因此我们可以借助别人训练好的模型，在国美互联网商品库中再次训练并进行精调，使模型达到应用所

需的要求。

## 未来的挑战

目前，国美大数据中心正在进行国美深度学习云平台（Gome CloudDL）的开发，使之能够支持各业务线在风控、图像分类、自然语言处理、人脸识别、推荐、搜索、广告等场景下的需求。Gome CloudDL 基于 TensorFlow 和 Docker 搭建，能够实现多任务资源的隔离、基于 Kubernetes 的调度、模型的持久化存储、TensorFlow 与 Spark 无缝集成等功能。

拍照购项目会继续在图像分割、实拍图处理、深度学习模型压缩、分布式搜索等方面进行优化，提高召回精度。

除了在召回模型中使用 CNN，我们接下来会尝试利用 RNN 来“捕捉”用户在点击序列中的模式，即利用用户点击行为发生先后顺序进行推荐和搜索的展示排序。同时还会进行深度强化学习(Deep Reinforcement Learning)的算法开发，根据用户所处的场景，在交互的过程中动态地推荐商品、活动、主题等，将传统的“商品推荐”升级成“场景式推荐”。

国美互联网大数据中心如同人工智能技术在国美的一片试验田，还有更多创新项目将在这里孵化。未来道阻且长，但探索的脚步不会停下。

**杨骥**，国美互联网大数据中心副总监。毕业于中国传媒大学并获得博士学位，博士阶段研究方向主要是计算机视觉和机器学习，包括图像的目标识别和语义分割。先后任职于凡客、京东，曾是京东 PC 首页与 APP 首页个性化推荐的开发者。多年来致力于个性化推荐系统与算法的研究和实践，目前专注于社交电商领域的深度学习技术。

# 刘文志：并行计算的未来， 是让人工智能无处不在

作者 杨旸



大数据、算法和计算能力决定了人工智能的发展。在计算领域上，主要依靠的硬件就是 GPU、CPU，以及最近刚推出的 TPU，背后是英伟达、英特尔和谷歌的角力。伴随着这些公司的股价一路上涨的趋势，从此也能看出并行计算的再次崛起。InfoQ 一直很关注并行计算领域，联系了业界领军人物刘文志老师，并拜托杨旸进行了一次较深入的访谈，以下是访谈实录。

## 并行计算和发展背景

**InfoQ：刘老师您好，非常感谢您接受我们 InfoQ 的采访。您在并行计算里的经验丰富，经历过图像处理、数据挖掘和石油勘探，能否介绍一下您所经历有趣的项目，和这些年并行计算在国内的发展？**

**刘文志：**我的工作经历，或者说我对这个行业的阅历大致也反映了并行计算在国内的发展。首先在我读书的时候，并行计算基本上是很少很

少有研究所（去做），或者是一些非常顶尖的高校才去做的事情。国内在 08 年之后 GPU 开始兴起，叫 GPU 计算，在一些科学计算领域开始得到应用。我也是在那个时候开始接触到这个，当时去做了更多的偏学术上（的应用），比如分子动力学，一些图像处理等等，用 GPU 可以比这种传统的 CPU 获得更好的速度。而在这种分子动力学这类上速度是非常重要的，因为模拟有时候是几个月，再快通过也常常要几周，用 GPU 我们可以



把它缩短到一周，甚至一周以内。那么这个时候就对发论文什么非常重要，因为我们在短的时间里多发论文，很快去实验更多的场景，可以得到更好的结果，而这对发论文都很有帮助。

我在11年从毕业的时候我就遇到一个问题，就是我未来要去做什么，当时去在各个网站上去搜这种相关的职位，基本上一个都找不到。所以这个行业只能去像NVIDIA这种公司。当然也很荣幸，我后来就去了NVIDIA，呆了三年，大致的工作也是计算机识别，石油勘探这些，也是从那个时候开始接触到深度学习。那个时候还比较原始，主要工作是去帮潜在的客户比如BAT等等，把应用的性能做好。那时就已经意味着国内已经开始有公司在往这个方向发展。而且也渐渐的发现，能够在招聘网站上能看到一些相关的职位了。在14年，我加入了百度IDL，那个时候已经开始大力的去招聘这个方向的人。从那个时候基本上可以看到，人工智能加并行计算和GPU计算这类，在未来会有非常大的应用，非常大广阔的领

域。但是整个的业界其实对这方面的人才的储备，包括高校都没有准备，甚至现在很多高校自己没有老师能够去做这方面的授课或这种教育等等。

整体来说，这些年，因为GPU计算，包括计算机视觉的发展，或者人工智能的发展，并行计算在国内的发展一路上升，而且是从产业界往学术界扩张的方式。所以计算机视觉加并行计算，或者说人工智能加并行计算，在未来还是会有非常大的发展。另外对工作来说是由市场的需求导致的，所以包括BAT以及其他大公司，对这方面人才需求也非常大，给的薪水也相对来说比较高。

在个人的项目中，让我比较感触，现在依然记得非常深刻，更多的是做一些石油勘探、数据挖掘和计算机视觉优化的项目。我们当初在百度IDL做了一个CNN的中间的卷积步骤优化，当时性能最好的CNN的库还没出来，CuDNN也没出来，要由计算机视觉的研究人员自己写代码，我们当时觉得性能最好的，最流行的框架叫CUDA ConvNet，GPU代码做得非常棒，它的作者是

Alex Krizhevsky, AlexNet 的创造者。我当时做了一个工作，我们发现它在新 GPU 上的性能并不如预期，我去在 NVIDIA 的 K20 上对它做了一些优化，可以达到当时那个 ALEX 原来的版本的性能的两倍，当然后来我们做出来大约一年后 Alex 也升级了，也跟我们当时的性能差不多，我们后来把这个成果反映给了 NVIDIA，然后相关的东西应该也已经被 CuDNN 吸收了。

**InfoQ：挺好的，我们自己关注也是感觉好像 2010 年、2011 年这个时间段，实际上是 GPU 开始在通用的计算里头做了一些尝试，美国好像有一个公司做 GPU Database，叫 MapD，也是从学术，他们做 Paper 的时候觉得速度特别慢，用 GPU 来做计算。**

**刘文志：**数据库本身可能并不是一个非常适合拿 GPU 来做，因为数据库大多数运算是增删改查，计算量相对来说比较小。

**InfoQ：它其实并没有很多的迭代，数据库比如 select 这样的，都是非常简单的一些计算。我看到您说这个当时就是实际上是你自己在 CUDA 上头做了一些，是 AlexNet 的一个系统上做了一个优化是吧？**

**刘文志：**AlexNet 是一个深度学习神经的一个网络，也是早期非常常见的，现在也有很多公司在使用它做 Benchmark。

**InfoQ：当时是 CUDA 已经出来了，你们在 C 的层面上头做了一些代码上的修改？**

**刘文志：**我们在 C 的层面上对卷积运算做了一些优化，性能超越当时 Alex 所做的 ConvNet 库的性能，然后我们当时也发了一个专利。

**InfoQ：那所以说您还是在 CUDA 的 kernel 和数据交换上做了一些修改工作是吧？**

**刘文志：**在 kernel 层，我们使用了一种和

Alex 不同的计算方法，这个计算方法在当时新的 GPU 上性能更好。

**InfoQ：很多最开始的时候，确实是大家需要在 kernel 层面做很多运算的优化，当时是 kernel 并行执行上的问题，还是 Host Mem 和 Device Mem 数据交换造成的这种瓶颈呢，我自己比较好奇。**

**刘文志：**当时主要是计算和缓存交换之间的同步没有做好，在时间上没有充分地相互交错掩盖。

**InfoQ：确实是，您说的这个是很典型的 CUDA 里面的优化内容，实际上要把数据交换时间用 Kernel 计算 hidden 起来。同步的时候，数据准备和 kernel 执行的 timing 要协调好。**

**刘文志：**对，就是避免他们一个在运算，另外一个再等着。

## 商汤人工智能应用场景和技术

**InfoQ：第二个问题，回到商汤现在这款产品上面，你能不能举几个现在市场需求比较大的智能识别场景，跟我们分享一下？**

**刘文志：**常见的一个一个是安防监控。安防监控对于人类来说是一个强需求，因为安全我们都非常重视，不希望自己回家时候发现家里已经被光顾了。更具体地，对公安来说，他们更希望能够找到嫌疑人。人最明显的标志是脸，因为人脸特征非常丰富，相对来说比较容易识别，相对来说比指纹更容易在公共的场合获取，所以通过人脸识别来抓到嫌疑人或找到嫌疑人，在公安领域非常受欢迎。

在另外一些场景下，我们需要知道某些人是不是带了违禁物品。比如一些恶性事件里，我们需要在嫌犯拿出刀、枪的时候，检测到，及时向

周围巡警发出通知。这样，我们在安防监控上可以让人们的生活变得更安全。另外，在日常的生活中，比如最近的手机卡实名制。之前因为历史的原因，中国大多数人包括我自己都是不太喜欢购买手机卡时提供名字等个人信息，所以通常用假名字。这个政策出来后，可能要得去中国移动营业网点重新去登记一下。对于我个人来说可能不是什么大问题，打个车来回一趟不麻烦，但是对中国移动来说，是个非常大的需求，因为中国可能有 10 亿个左右类似情况的手机号，如果为这 10 亿个人办的话，要在几个月内完成几乎不可能，用人脸识别就可以帮他来解决这个问题。比如我在家里可以打开一个中国移动的 APP，把我的个人信息，身份证件和我的脸做适配，把我的个人信息和手机号定义起来，可以减少移动这方面的工作，对移动是非常重要的。

另外如互联网金融：如果事先在家里能处理好开户和某些业务，能省下去银行的时间，省了两个小时的排队。在网上自己处理可能半小时，或几分钟就搞定了。

还有一些偏娱乐的场景，也是商汤占有优势的应用场景。比如直播，美女通常都希望自己的镜头里面表现的非常漂亮，那我们如何在直播里面，通过人脸技术去提高网红颜值，也非常重要。

**InfoQ：您谈到这个，互联网金融和安防监控确实是行业里头谈的比较多的，这两个应用已经有几年的时间了，那么您刚刚谈到这个网红确实以前很少关注过，娱乐行业也好，现在对人脸识别的实时要求也挺高的。**

**刘文志：**娱乐也是人们的日常需求，通过我们的人脸技术，甚至与其他技术结合，可以把美颜等做得更好，把人的外观体验也做得更好。另外比如两个人在聊天时候，可能对方觉得我现在

穿着睡衣不太合适，就把我的衣服换成一个西装等等。

**InfoQ：**是，现在对直播的监管要求也比较高，而且如果你纯粹把它看成一个互动娱乐也挺有意思，比如我想让主播换个装什么的，真的可以虚拟实现。您说的这些场景一部分是识别，一部分需要结合实时分析。那咱们是围绕图像识别和人脸识别提供引擎，还是提供一个完整的系统，包括图像识别，也包括了实时分析，以及消息系统？

**刘文志：**这些商汤都提供。一方面，我们提供这种偏 IP 的 SDK，另外一方面，也提供整个业务系统，比如安防监控服务器，我们可以提供整个服务器给客户，供客户直接使用，不用二次开发。

**InfoQ：咱们提供的接口是什么样的呢？**

**刘文志：**这个关系不大。对于大多数工程师来说，C 肯定是一个必须要掌握的编程语言，其他语言基本都能加入 C 的支持，所以呢，我们要提供一个 C 的接口，去包装 Android，包装 Python 接口。基本上所有用户都可以接受 C 的接口。

**InfoQ：现在用深度学习实现的人脸识别也有一些不同的技术，一些开源框架，像 kaggle 网站上也有一些分享和实验，商汤在人脸识别，图像识别上的优势是在什么地方呢？**

**刘文志：**如果单从人脸识别的技术上来讲，业界的几个顶尖的公司的差别不是非常大。但是对整个产业来说，优势从来都不是比单个点。更多的是综合集成的优势。

对商汤来说，从技术到算法，到平台，我们都有优势。我们有自己的做深度学习的框架，相



比与Caffe、TensorFlow，我们有自己的一些特色。当然这些特色如果跟TensorFlow一比一的比较，可能各有优势。但是好处就在于商汤可以完全控制该框架，我们可以按需求去做修改，对特殊场景去做非常好的优化，而这很难通过TensorFlow框架获得的。另外，我们在算法上面，有非常多的研究人员，技术上可以开发非常好的技术版本，超越业界的顶尖水平，另外，也可以关注新领域，新方向，去看未来一年两年可能在哪些领域上取得突破，预先做准备。

除了在训练平台上保证我们技术领先之外，在部署和交付用户时，如何保证也是领先的，这样是一个非常大的挑战，因为客户需求是千变万化的，有的需要在云上部署一个应用，有的可能需要在手机上部署一个应用，有的客户可能需要在摄像头部署，有的客户可能需要在手表上部署。在这些千变万化的应用中，我们需要有一个平台，一个技术框架来保证我们的算法都在各个平台上能够跑到比较好的性能。因为在云上一般

是用X86，计算能力在一百G flops左右，在手机单核通常就十几G，嵌入可能就几G，差别通常几倍到几十倍。我们在云上跑的很好的算法，可能在手机上跑得非常慢，嵌入式可能就跑得更慢了。

那我们怎么样把不同平台的性能做好，我们商汤自己开发一整套系统，从云到手机，到嵌入式，甚至到FPGA，我们都做了很多工作，让我们和竞争对手拉开很大距离。

另外，在数据上，因为我们商汤从成立一开始，我们就非常关注数据的价值，我们通过公开渠道获取了许多重要的人脸数据，做了标注等工作，人力投入也是非常大的。所以单讲算法优势，我们可能并不是非常大，但是，把集成其他的优势，我们相比竞争对手，或者其他一些开源技术，是非常大的。

**InfoQ：谢谢。您刚提到商汤自己开发的深度学习框架，如果我们的开发者想尝试一下试用，需要准备怎么样的开发平台呢？**



**刘文志：**目前来说，如果要把我们的框架公布出去，会带来另外一个需求，很多同事，朋友会好心地给我们提需求，但按照我们现在的体量和现在面临的压力，没有时间和精力去做这个事情的。目前来说我们不太会公布。更多的是把这些东西在我们自己的产品上使用好。

关于如何提升一个平台来去做深度学习的训练和部署，国内有许许多多的厂商已经非常的专业了，像浪潮，曙光，AMS 等等，他们都已经做了很多年的积累，对这方面的需求也比较了解，可能更熟悉。我的经验在于，如果需要一个好的深度学习的训练平台的话，我们可以去基于 NVIDIA GPU 自己去搭一个。通常关注的问题主要是两个，一个是整体的功耗，整体的功耗限制了我们要用多大的电源，机房线路能不能承受的起。另外要考虑空调散热和噪音，通常单机的话不会存在这个问题，但是如果是一个小机房就要考虑空调的问题。

**InfoQ：**您谈得有道理，如果要维护一个深

**度学习的开发框架，所需要付出的精力不是一点点，很多人会提很多的需求。**

**刘文志：**这也是现在为什么我们现在并没有公开，如果有一天发展得像 BAT 发展的这么大了，我们也很愿意把它公开出去，因为那个时候我们就有人力来维护这个事情，而且这个事情对整个产业都是非常有利的。

**InfoQ：**您刚刚提到在不同平台上头做部署，在手机端、摄像头、嵌入式上的部署。NVIDIA 也是有这样的考虑，在嵌入式设备，在云端，在高性能计算也需要兼容不同的 GPU。是不是也是类似于这样的框架思路，在不同的设备上的 kernel、代码是一样的，只是编译到不同的设备平台？

**刘文志：**因为不同的设备有不同的特性，计算能力不一样，希望有共同的模型，思路基本上是一致的。

## GPU编程应用与深度学习

**InfoQ：在并行计算上，英伟达和 Intel 在国内都做了很多工作，nvidia 进来得很早，英伟达近年发展也很强，我们知道天河二号用了 Intel 的 XeonPhi，英伟达各个系列的 GPU 的开发者群体很大，您有没有一些心得，或者观察到的一些有意思的东西跟我们分享一下？**

**刘文志：**从面向市场的宣传来说，GPU 比 XeonPhi 更早。Xeon Phi 当时的目的之一是 Intel 为了避免在 HPC（高性能计算）里被 Nvidia 超越。但是在深度学习领域，都是为了面对深度学习训练的需求，目标是一样的。不同点主要两个方面，一是生态系统，二是芯片的设计。Xeon Phi 相比于原先的 X86 的架构做了一些更适合做众核的改变和调整，而 GPU 本身擅长图像处理、渲染，运算有非常大的并行性，更关心整幅图，整个场景，最终渲染的时间，而不是说某个像素的结果，设计时有非常大的差别。

在生态系统，Intel X86 坚持的是串行优先，兼顾并行的方式，构建生态系统的思路也不太一样。

但是整体来说，我个人看来，无论是 Intel，还是 Nvidia，他们都很难单独把人工智能这个市场吃掉。站在消费者的角度来说他们的竞争是好事，一方面能够提高整个行业的技术水平，给消费者提供更好的产品和服务，同时会促进整个市场的繁荣和行业的向前发展。

**InfoQ：如您提到的，Xeon Phi 也好，GPU 也好，在并行计算上，思路有一定的区别。比如 GPU 以计算单元比较多，而控制流转、控制单元和缓存都大大减少，不同处理器的架构对编程思路也会有区别，像您说的话，GPU一幅图，是一个单向逻辑，多个线程同时做一个，而不用很多跳转。这对开发者和代码维护来说有很大挑**

战的。如果你一旦你的程序是面向 GPU 写的话，如果今后想转移到 Xeon Phi 上，代码调整还是很大的一个工作量，两种处理器上的编程不完全是兼容的。

**刘文志：**如果要深入细节的话差别是有。你刚才提到的那个关于控制的方面，在 GPU 上是要减少控制的比例，在 Xeon Phi 也是同样的，因为控制它本身会导致这个并行的时候，一些工作是浪费掉的。而且控制会导致 Cache Miss，分支预测失败等等。从优化的角度来说是近似的，只是说优化的技术在哪个架构上产生的收益哪些会大些，哪些会小些。

**InfoQ：这个话题可能有点走题，不过前段时间，我也关注一下并行计算优化层面的东西，包括编译器层面的优化，实际上现在也有很多挑战。并不是完全有定论，所以我们这个时代好像很有意思，正好在一个快速发展的阶段。**

**刘文志：**是在快速发展。技术百花齐放，多分支，碎片化的发展等等。

**InfoQ：现在 NVIDIA 的 CuDNN 用来加速 Caffe，TensorFlow 和 CNTK、Theano 这样的 Deep Learning 的框架实际上越来越多，我不知道您在工作里是什么时候开始接触到 CuDNN？您在英伟达工作过，CuDNN 从什么时候开始在中国有明确的需求，最开始的需求是什么样的，你对这个技术本身有些什么样的评价吗？**

**刘文志：**CuDNN 我接触最早是在 14 年，NVIDIA 第一次发布的时候，我在百度 IDL 就开始使用，它做得非常好的地方是给大家减少了我上面提到的自行对 ConvNet 的卷积计算优化之类的工作。而且 Nvidia 通过这种方式一下子把整个大家用 GPU 做深度学习的性能提高了。原来我

们可能需要两个 GPU 才能做完，用 CuDNN 之后，可能一个 GPU 就搞定了。好处之二是，很多时候我们不用自己去写 framework，因为 Caffe，TensorFlow 之类都支持它，那么直接可以按自己的喜好选择这些框架来用，性能区别不会太大，可以忽略。因此对于用户来说，就更容易根据自己的喜好，根据自己的习惯去选择，而不是根据计算速度去选择。CuDNN 本身是非常底层的，在使用的时候需要知道很多细节的库，那么在使用的时候，我们需要非常注意它的各方面配置。尤其是卷积，它有不同的算法实现。要注意各个算法的性能和其他指标的搭配。比如某些场景下，我们希望速度尽量快，在另外场景下，我们希望使用的显存尽量的少，还有些情况下，我们有些其他的需求，比如我要保证计算的结果跟某个计算的结果是一致的，这时，我们要非常注意它使用的具体算法是什么。

**InfoQ：我记得 CuDNN 的介绍里也谈到，同样目的的具体算法可能有三四种，每一种在计算、内存、涉及到的数据交换，处理好像不太一样。所以开发者还是需要了解这些细节。CuDNN 有没有提供某一些监测界面，来看运行状态？比如运行时，开发者用某些工具查看不同的几种类型的运行情况，看看目前所用的类型的运算情况是否合乎自己的期望，不行的话，就换另外一种？这个可行吗？**

**刘文志：**这个可行，在 CuDNN 里面有些内置就有这个功能，当然我们需要根据自己的具体情况，把定的参数输进去，这样我们就可以知道每个场景下的这些数据是什么样的。在大部分场景下，更多希望性能最优，所以我们用运行时的查找的方法就可以了。

**InfoQ：现在 Deep learning 的框架蛮多**

**的，一些开发者对多 GPU 并行比较感兴趣。好像并不是所有框架都支持多 GPU 并行，比如 CNTK，可以通过 Microsoft 一些特殊算法执行可以支持，但是像 Caffe，TensorFlow 似乎还不是。您觉得哪些框架对多颗 GPU 运行的支持相对比较好一些？**

**刘文志：**整体上来说，Caffe，TensorFlow 都支持多个 GPU 的并行，Theano 我不太清楚，Theano 本身更偏学术一点，希望用脚本语言来处理，灵活性非常高，但是是否支持多 GPU，目前我不是太了解。

Caffe 和 TensorFlow 都支持。有一个 NVIDIA Caffe，是经过 NVIDIA 优化的，支持单机多卡。Caffe2 可以支持多机多卡，TensorFlow 既可以支持单机多卡，也支持多机多卡。我们的测试表明在 NVIDIA Caffe 上，单机多卡的性能还是挺不错的。TensorFlow 的单机多卡的性能，最近的版本还行，但多机多卡的版本的性能相对来说比较一般。我估计一个原因是因为谷歌更多关注自己的应用，很有可能 TensorFlow 在谷歌的数据中心里面性能比较好，但是它的数据中心的情况，跟你我用的不太一样，可能 TensorFlow 就不太适合你我的数据中心。这也是为什么我们自己去建数据中心，自己去开发深度学习训练框架的原因。因为我们的算法不一样，那我们的数据中心就要相应调整，我们的框架也要做调整。从这角度上来看，我觉得，任何深度学习，人工智能公司只要想做大，那么深度学习的框架最终都必须自己去做，依赖于开源的，或者说依赖于商业的解决方案并不是很好，也并不是很合适。

**InfoQ：您说的开源的问题我们也有同感，借助开源的框架或 Package 来起步和早期实验没问题，但是要把一个产品做强，推向市场，恐**

**怕一方面要自己掌握核心技术，同时参与这种开源社区就要有 committer 级别的人。**

**另一个问题：一些应用里用单个 GPU，或者一个单机里有多颗 GPU 能做很多事情了，那么现在的情况下，到底有多必要，把从单机版推向多机多颗 GPU 并行？**

**刘文志：**这个答案是非常肯定的。这么说可能有点过：如果用不到多个数据中心来做深度学习的公司绝对不是最顶级的。道理很简单，对深度学习来说，我们有许许多多的应用和配置需要优化，但是我们如何去选择一个更好的配置，这需要去实验，去做判断。如果我们有一个大的数据中心的话，那实验的次数、频率、结果和配置就会非常多，就意味着，我们可以从中更好地了解这个应用场景的特性，获得更好的模型。所以如果要在这个领域做得好，一个大的数据中心，一个好的框架是必不可少的。

**InfoQ：您谈到这个也是现在多 GPU 的一个典型场景：一种情况是把一组 GPU 都用同一种模型，而数据集不同；另一种是在不同的 GPU 上用同样数据训练不同的模型，得出一个优化结果。您刚才是说，是在训练阶段，来尝试不同模型呢，还是说在客户的现场，部署下去以后，用多颗 GPU 实现线性拓展，缩短判断时间？**

**刘文志：**这个看用户需求，大部分情况下，在部署的时候，单个 GPU 来跑应用可能更好，而在训练的时候不太一样。

**InfoQ：恩，训练和部署还是有一些具体的区别。让我们回到 Xeon Phi 上，听说以前的一些代码略微做一些修改，就可以在 Xeon Phi 上实行并行化和加速。这很有现实意义——我们以前开发的串行代码很容易移到这样的架构上面来。这种将以前代码并行化移植到多核环境，是**

**不是在不同的处理器上头确实有不同，是不是像 Xeon Phi 这样的，可移植性更强一些，并行化更容易一些？**

**刘文志：**商汤现在并没有用 Xeon Phi，但是我个人在 Xeon Phi 上有些开发的经验。任何想要拿到 free lunch 的想法都是徒劳的，从大的层次上来说，用 Xeon Phi 来实现并行化和加速，希望不做修改不现实。不修改代码，在新的处理器上运行不难，但是，如果将一个 X86 上的程序移植到 Xeon Phi 的 60 个核上，希望获得 30 倍的加速，仅仅靠简单地修改一下，或者说重新编译一下，是不现实的。

要获得对应的加速，要付出相当多的时间。对于深度学习的公司，短期内换到 Xeon Phi 上不太现实，因为今天所有的关于深度学习的框架，都对 GPU 非常好的支持，但是对 XeonPhi 的支持很少。只有 Intel 自己的 Caffe 对 Xeon Phi 提供了支持，而 TensorFlow 和其他的框架没有足够支持。在这样情况下，切换到 Xeon Phi 上，可能为时过早。另外，也需要整个生态上对它的支持非常好，才有可能有让深度学习的科技公司考虑迁移到 Xeon Phi 上。现实情况是 Intel 并没有把它的对深度学习和训练的投资集中在 Xeon Phi 上面，还看不到这样的保证。

**InfoQ：我能理解，像卷积计算在 GPU 这种多核并行的架构上很有优势。如果单纯从硬件加速角度来看，为了利用多核来把以前的应用，转移到一个多核平台上，Intel 等厂家推出了 Cilk, OpenMP 这样的框架，通过加入一些 directive，就可以把部分执行编译成并行的，这样，看起来对代码本身，对逻辑本身的修改相对较少一些，也不需要在编程初期考虑并行或处理器 SIMD。但是如果通过 directive 这样的方**

**法，也许能提高一些性能，但是能加速多少？并行度能提高多少？对后面的优化有怎样的影响？我们心里还是没底，你在以前的工作有没有接触过这种加速？**

**刘文志：**这个我用的比较多了，但是整体上来说，像 OpenMP, Cilk 等还是多核级的并行工具，并不是众核级的并行工具。当应用的并行度不太高，每个的粒度也比较大，那么用这种多核级的 OpenMP 可能会获得比较好的性能，但是一旦并行度非常大，OpenMP 和 Cilk 这种粗粒度的就不太适合了。因为当并行度比较低的时候，只要几个线程就够了，但是当并行度非常高的时候，可能要有成千上万个线程甚至更多，这种情况下，这些线程之间的交互，相互影响如何控制？这在 OpenMP 里面没有多少支持，相对比较简单的支持是通过 Synchronize，或者说同步，原子函数等，但是这些同步的代价又非常的大。所以本质上从我的经验来看，OpenMP 这类还是多核的编程方法，并不是众核的（如 GPU 或 Xeon Phi）。Intel 用 OpenMP，更多的是想让之前开发的多核程序能很快地迁移到 Xeon Phi 上，获得加速，但是实际的性能结果不会像预期的那样好。

**InfoQ：您说的也对，NVIDIA 在进程通信、在 Barrier 和同步机制上面，或者 Share/Global 显存，mapping，pinned memory 上都提供了很多更细腻的手段。商汤下阶段在哪些方面会重点开发，下阶段可以看到哪些重点应用和重点方向呢？**

**刘文志：**从我的人工智能加嵌入式并行计算的领域来看，嵌入式，或者 IoT 是未来的方向。可以分为几个大的应用场景。一个是云；一个更靠近人的城市或社区；再进一步到每个人身边和身上。对于云，整个业界已经关注的非常多，

目前看起来做得最完善；城市和社区已经有许多安防公司在做。但家庭和个人上还是待开发的领域。

真正要把人工智能做到无处不在，只在云和城市是不够的，最终要落实到人和家庭上，而 IoT 就是这样的技术：怎么把 IoT 和人工智能结合起来，这是未来的方向。

在我加入商汤之后，也就大约两年前，就把我在所在部门的使命定义为：让人工智能无处不在。这个观点最近已经被许多的公司，包括 BAT 的一些部门，也定义成自己的使命，让我很欣慰。我相信让人工智能无处不在的 IoT 和嵌入式是未来的方向，而且在嵌入式的人工智能上创业和扩张的公司将会越来越多。甚至从某种程度上人工智能的新的独角兽将会出现在嵌入式领域。

**InfoQ：能不能理解成家庭的安保、家庭的摄像头，反侵入？**

**刘文志：**反侵入，感知。在家庭里通过人工智能，能做许许多多的工作，家庭安防，家庭辅助是很小的一方面。大的一方面是机器人，我相信最终每个人的家庭都会至少有一个机器人，无论是扫地的，还是家庭辅助的等等。而机器人必须需要嵌入式和人工智能。

**InfoQ：这很有意思。现在像家用机器人，工业机器人，安防都是大数据领域这几年大家看好的一些发展方向。那您觉得机器人在工业制造业里，现在是不是有很好的一些发展方向可以做，还是说前几年的工业机器人视觉的技术积累，已经相对可以满足现在需求了？还看不到下一个更能可能突破的方向呢？**

**刘文志：**我个人对工业机器人并不是太熟，我看得到的，有很多事情通过传统机器视觉的方法在机器人应用上做得还不够好。在我们可以看

得到，大家认为相对比较好的机器人的创业公司，比如波士顿动力的机器人，离人类的期望还有差距。这个差距怎样通过人工智能和拟人来弥补，其实是一个很大方向。简单的来说，我们怎么让机器人在家里抓起一个物品的时候，感觉是人在抓起一个物体。它在跟我们握手的时候，我们就可以感觉是一个人在跟我们握手，而不是机器，这需要很人工智能的技术，比如拟人的技术来辅助。我是觉得一旦这个技术做好了，机器人真的会进入每个人的家庭。

**InfoQ：所以说这实际上需要按视觉，分析物体的重量和形状，细致调整手上的力道的一个闭环的系统，我能理解。工业机器人的发展受限也是跟工业生产线上的那些东西都很固定，零件基本上就是那些东西，对识别之后的行为要求不是很高，它用传统的特征提取，一个很固定的模型就够了，不需要太多学习？**

**刘文志：**对。但是未来的机器人不是这样的，尤其是到了家庭，到了社区的时候。

**InfoQ：在一个比较自由、开放的环境里进行识别，形成动作，就完全不一样了。**

## 对新手的建议

**InfoQ：我们 InfoQ 的读者现在很多都对 AI 比较感兴趣，或者以前从事相关行业，希望在自己工作中使用 GPU 或深度学习。对这些刚起步的一些朋友，你觉得有什么样的建议，怎么样让他们能更快的学会一些干货呢？**

**刘文志：**整体上来说，今天的人工智能、深度学习、机器人的发展已经远远超过了三四年前。今天已经非常容易学习各种框架了。

首先框架现在已经非常多，很多都是开源的，对初学者来说可以很容易获取到它们。硬件，

GPU 的性能已经非常高了，价格也相对来说比较便宜，很容易买到。

但是在具体的步伐上，我建议大家可以通过这些步骤，首先可以花一两周的时间去学习一下深度学习的基本理论，比如说吴恩达在 Udacity 上的视频，斯坦福有一个网上课程等等，通过一两周的时间理解理论基础和一些基本认知，然后在具体的工作的时候可以去重新去加深，去查找。另外，一两周之后，可以找一个常用的深度学习框架，比如装个 Caffe，TensorFlow，跑一些例子，了解一下它的运行过程。完成这步之后，如果还愿意去用深度学习做些事情的时候，可以组装一台 GPU 的服务器，当然这个价格就可能比较贵了，可能达到十万左右。当然如果只是个人学习，可以买一台高配置的 NVIDIA 的游戏配置的台式机。然后可以把深度学习的框架，和 CUDA 环境等等配置好，用来跑程序，去看具体的代码实现。比如一个新的任务，在原来的 TensorFlow 或 Caffe 上面都不支持，应该如何去修改这个框架，才能让它支持，这基本上就可以来比较好地完成一些工作了。我个人并不建议大家重头去搭深度学习的硬件平台，因为这不是核心，也不是重点，深度学习在现在已经这么流行了，各个厂家，各个硬件的集成商都有他们的配置，我们直接按预算去购买一个合适的配置是更好的选择，把时间花在学习上更有价值。

**InfoQ：我们最近关注到 NVIDIA 之类的公司推了一些比较简单的入门产品，比如说 DIGITS，微软在 Azure 上有 Machine Learning Studio，用拖拽可以搭一个神经网络，用一个样本数据集来训练。这样的东西你觉得对于哪种阶段的朋友会有帮助，而对哪些阶段的朋友不适合？**

**刘文志：**我建议刚开始学的时候，开始花一两天时间去学习一下是可以的，但是如果想要去把东西做得很好的话，不应该去使用这个。把实现的细节最终隐藏起来，能方便最终用户，但是对于希望把技术做到很好的同学来说，这个是反而阻碍他的技术发展，因为从来就没有免费的午餐。要把东西做好就必须要去了解细节，比如我们去改东西的时候，如果我们只会拖拉，是不可能改好的。如果我们只会去跑一个 Demo，我们要去改它里面的细节也是不可能的。所以建议在早期的时候，可以去参考这些东西。另外，如果我们确信，要用的东西是非常固定的，不会有什么大的变化，用了这些东西也可以满足需求。如果想要做得非常好，如果有拓展的需求，那么还是不应该去使用这些东西。

**InfoQ：**那么还有哪些开源的包，或者是一些常用的 library，你觉得哪些作为起步阶段，可以先下载，用一些 Python 脚本来做一些很简单的神经网络的尝试，有哪些开源工具会比较好呢？

**刘文志：**一个是 Nvidia 的一个库：TensorRT，用来部署。如果我们要把东西交给客户，让它在 GPU 上运行的时候，这很容易上手。其他还有一个工具叫 DeepLearning4j 或 DL4J，是一个 Java 平台上的 Deep Learning 框架，现在安卓、和云的程序员还是非常多，可以去看一下。因为使用的方式跟 Java 理念比较一致。

**InfoQ：**非常棒，非常感谢您的时间，我自己的收获很大，我相信我们的读者收获也会很大。

**刘文志（花名风辰），**商汤科技高性能计算总监，毕业于中国科学院研究生院，闻名于并行计算江湖，尤善异构并行计算（X86、ARM、GPU、APU 及 PHI）和大规模集群计算，有 7 年相关经验，涉及图像处理、计算机视觉、数据挖掘和石油勘探。曾任英伟达并行计算工程师（协助建立英伟达北京 CUDA 团队）、百度在线高级研发工程师（协助建立百度深度学习实验室异构计算团队）。著有《并行算法设计与性能优化》《并行编程方法与优化实践》《科学计算与企业级应用的并行优化》《OpenCL 异构并行计算》四本并行计算领域的专著。

# QQ空间亿万图片存储架构的发展四阶段

作者 付强



QQ 空间 (Qzone) 是腾讯公司于 2005 年开发出来的个性化空间，12 年过去，目前 QQ 空间用户数量达到 6.233 亿。其中，QQ 空间背后的平台照片总数接近 1 万亿，在这背后的存储架构是怎样的？

## QQ空间后台开发团队的背后协作

首先先介绍我们 QQ 空间后台开发团队，QQ 空间相册后台整条链路相关的团队有 5 个：

负责相册业务开发与整体优化的相册后台团队；

- 负责图片处理的存储开发团队；
- 负责底层结构化数据库开发的数据开发小组；
- 负责冷数据备份研究的一个小组；

- 负责业务运维与机器管理的运维团队

除此之外我们对接对接客户端 IOS, AND, 前端 H5 与 WEB 开发的各有一个小组，对接具体项目的推进。

目前我们团队主要负责相册的架构优化，整个系统的性能提升，成本优化，产品特性开发与相册历史数据的挖掘以及推动机器学习在相册照片相关场景的探索与应用。

我们不同技术团队的分工与配合主要有两种

方式：

- 一种是具体项目合作的方式。这种一般是目标比较明确的具体项目推进，各个团队一起参与进来，会有具体的项目负责人和项目经理进行职能的分工与项目进度管理。
- 另外一种是基于各小组的职能划分，做一些长线探索与技术预研。比如存储方面的深入预研，新型图片格式的研究与推广，低成本的数据备份技术，统一运营管理平台等。很多技术成果同时复用与推广到内部产品比如微信，微云，邮箱，腾讯云上。

在腾讯内部，只要你目标比较明确并且有足够的挑战，这方面的推动合作往往比较顺利。比如我们上半年与音视频合作的采用 TPG 图片流量优化项目，涉及 4 个部门的多个团队。整个移动端流量优化掉 60%。对于 GIF 格式的图片的节省流量 90% 以上。

## 回顾12年老产品的开发之道

QQ 空间已经有 12 年的“老”产品了，经过 12 年的反复迭代与演化，可以说空间是目前最复杂的产品之一。因为涉及业务广、入口与场景多、历史包袱重，所以一度我们饱受开发效率和维护代价的困扰。目前针对这个问题我们的做法如下所述。

### 第一，多从产品层面切入与优化，包括历史逻辑（业务）的清理与下线。

同时对于新上的业务逻辑，在设计之初就充分考虑逻辑的解耦与下线逻辑内置。事实证明，基于产品层面业务逻辑的优化对于清理历史包袱有很直接的贡献，这一点往往很容易被开发者忽视。

### 第二，代码架构设计多考虑扩展性与功能复用。

这一部分我们的做法是多做一些底层的通用标准化组件框架，比如我们现有的 CGI 的组件、Server 开发的组件、移动端接入的组件等等，这样可以把很多的业务逻辑处理标准化起来，节省不少开发与维护成本。

即使这样，随着业务的不断变化，目前我们的代码还是存在很大的冗余与需要优化的地方，这里的矛盾是业务的重构与优化是无止境的，而对于业务开发来说，资源与时间都很紧缺。

这里我们的经验是把代码重构与需求开发结合起来，在我们的业务开发过程中附加上部分重构优化的工作，或者把架构优化跟成本优化结合起来，这样能更好地推进重构的进展。

## 平台照片接近1万亿的困境与架构优化的四个阶段

从用户层面考虑，随着业务规模的变大，意味着所面临的问题会随指数级放大。

对于业务规模比较小的系统，万分之一概率出现的系统问题可以不用太关注，但当业务规模达到万亿级别时，哪怕是百万分之一、千万分之一概率的问题也会因为这个系统规模进行极端放大，因为会影响到相当基数的用户而变成不得不优化解决的问题。

同时从业务层面考虑，你的体量起来了，这里的业务成本与运营代价也会变成不得不经常考虑的问题。

我们业务的第一次架构优化是从 MySQL 集群升级成自研的海量文件系统 TFS (Tencent File System)，支持用户数据规模可以达到 EB 级别，在这当时看来基本相当于整个系统的无限容量的

平等扩展了。这个阶段一举解决了容量问题，为业务量的爆发打下了基石。

我们优化的第二阶段主要集中解决用户的接入问题，包括极速上传、跨运营商接入、下载加速，优化后带来了用户较大体验的提升与数据量的大量增长。

随着新型存储介质的出现，以及用户面对照片容量与性能的要求，第三阶段我们的优化重点主要是进行 TFS 的重新设计，主要支持了单个用户的无限容量，同时采用了基于 SSD 介质（TSSD）和高速索引升级版的 TFS，同时对冷数据进行特殊处理以降低成本。

架构优化的第四阶段主要是针对移动端、弱网络进行优化。目前我们优化方向主要集中在高性能，低成本与更强的数据处理能力方面。

现有的系统从存储容量上来看，我们实际容量目前达到了 0.4EB，单从容量这一维度来看，目前的架构完全可以满足未来十年的发展。

在整个架构迭代过程中我们仍保持了高性能，一方面我们架构设计主要的设计思想是简单，包括业务逻辑的简化，与处理过程的简化；另一个点是我们目前采用的是结构化存储，能够针系统进行针对性的优化，这也对高性能有比较大的贡献。

最后我们会考虑整个系统全端优化，能够从用户端到终落地的存储各环节进行系统的优化来保证高性能。目前我们的存储架构也面临主要问题是结构化存储的灵活性比较差一些，这里我们也在借鉴业界的一些思想。

## 自研和开源

其实腾讯内部有很丰富的内部组件与开源软件。

存储来说，像基于 SSD 盘 Key-value 型海量存储系统 TSSD，内存与 SSD 混合型存储的 CKV，基于目录管理的海量文件系统 TFS 和冷存储系统 BTFS。

可以说，在主流互联网公司使用的开源组件在腾讯基本都可以找到对标的产品与解决方案。目前我们使用的组件自研的占比相对多些，主要原因是这些组件都有专门的内部维护团队，而且基本都经过了线上大规模业务的考验与优化，可靠性与实用性比较有保证。

同时我们工程中也有较多的使用开源组件，比如 Redis，Kafka，Ngnix，raft 等都有在我们工程中应用。目前一般独立的模块，中小型的系统我们比较倾向采用开源的框架，灵活性比较好。目前腾讯本身的思路也是尽量把内部好的组件开源出去。

## 为什么要引入机器学习？

机型学习目前的三大应用方向：图片（计算机视觉）、语音、文字。我们积累了超海量的有价值的图片数据。同时之前从我们部门发展出的优图实验室，在机器学习上面积累很深技术储备。最后，基于我们的产品形态、机器学习的成果能够很好的与业务场景结合并进行产品价值的最大化。

所以综合来说，我们具备“天时”、“地理”、“人和”三大因素，进行图片相关的 AI 结合的项目是比较自然的。上半年我们主要在做两个与图片相结合的项目：

第一个是使用机器学习来进行图片内容的识别与用户身份的识别，我们主要使用线上机器的空闲时段来进跑训练好的模型进行图片处理，目前应该是业界规模最大的图片处理系统，每天高

峰时处理图片上百亿。

另一个主要是利用客户端的 GPU 和 CPU 能力来进行图片的超分辨率的应用，这个项目也已经上线，能够充分把客户端的处理能力发掘出来，在保证清晰度的情况下节省用户 75% 的带宽。经过极致优化，能够在 80% 以上的手机上能把训练好的模型跑起来。

上述两个项目都取得了不错的效果。随着移动端性能的越来越强，包括现在 GPU 慢慢成为手机的标配，同时结合数据不断积累与算法研究不断突破，AI 相关的应用应该是未来的一个大方向，我们也希望在这一块有更多的发力。

## 存储的未来与建议

存储发展到今天，数据规模与性能目前来看已经不是最大的挑战。这两者目前来看应该都是走在业务前面的。目前比较有挑战的问题主要还

是集中在储存系统自带的数据加工处理能力，包括智能数据处理以及大规模存储对价格的要求。目前我们比较关注近几年出现的 NewSQL 和 JBOD，并且有望借鉴引入到我们系统中来。

作为后台开发 Leader，这些年最大的感受应该是不要把自己完全局限在后台这个领域。实际上前台相关的知识、业务相关的知识，还有通用的包括流程处理相关的知识能够很好地帮助你全面成长。

目前的分布式数据库架构很多，而且不断有新的架构涌现出来，如果要学习这方面的知识的话，建议结合自己的项目或者所做的工具，引入一种最好是开源的框架进行深入学习，最好是以应用过程中碰到的问题来驱动。**只要深入进行你会发现，其实很多高并发系统的设计思想是相通的。**

**付强**，腾讯 QQ 空间后台开发 Leader。负责国内最大的 UGC 图片平台架构设计与开发工作。

有着多年的海量系统设计与移动环境下优化经验，对分布式数据库架构，高可用高并发系统设计等方面都有较深刻的理解，目前平台照片总数接近 1 万亿。负责过多个技术优化项目，每年带来成本节省超过 1 亿元。目前致力于推动机器学习在图片领域的探索。

# 从实践者的角度看 Rust

作者 唐秀涛



**唐刘：**我个人并不是一个语言爱好者，也不会刻意的去研究不同的编程语言，但我的语言倒是蛮多的。我自认为应该是一个工程实践派，也就是要解决这个问题，哪种语言好，就用哪种。

我个人最喜欢的还是 Go 语言，无论从开发效率还是运行性能上考虑，Go 都是非常不错的选择。当然，如果现在要过于关注性能以及跟操作系统打交道，我会使用 Rust。

在工作中，我们主要使用 Rust 和 Go，毕竟整个 TiDB 都是基于这两门语言打造的。

编程语言相关的讨论，几乎是技术社区永恒的热点话题。在 QCon 上海 2017 上，我们也专门设置了《编程语言》专题。届时，PingCAP 首席架构师唐刘将分享《Futures and gRPC in Rust》。QCon 在会前采访了唐刘老师，交流了编程语言方面的一些问题。

**QCon：TiKV 选择用 Rust 实现，具体应用中感觉 Rust 有哪些地方体验比较好？**

**唐刘：**使用 Rust 的好处在于只要你挨过了

最初学习的阵痛期，写代码会非常的高效，写完只要通过编译了，几乎不用担心 data race, dead lock, dangling pointer 等问题。

另外，Rust 的包管理也做得很不错，在 crate 上面可以找到非常多的高质量第三方组件，我们团队也贡献了几个。

如果使用了 clippy，在编译的时候，Rust 还会告诉你这样写代码虽然是正确的，但最好改成这个样子，所以你看大家的 Rust 代码整体风

格会非常的一致，简洁美观。

### QCon：是不是遇到过什么坑呢？

**唐刈：**相比 Go, Rust 在 profile 上面的工具其实比较缺，虽然有 perf, systemtap 这些外部工具可以用，但也会给使用者带来新的学习负担。

Rust 对于网络程序的编写并不友好，之前就只有一个可怜的 mio，要实现个高效 RPC，几乎自己要做非常多的工作。譬如我们就自己基于 C gRPC 做了一个 Rust 版本的。

不过后面 tokio 这套生态起来了，情况可能会好一点。

### QCon：Rust 也在不断发展，你们会一直跟进新的版本吗？

**唐刈：**是的，我们在 TiKV 里面一直用的是 Rust 的 nightly 版本，然后会定期升级到最新的 nightly，我们在这个上面的策略还是比较激进的。

### QCon：团队是不是也对 Rust 的发展贡献了自己的想法？对 Rust 的未来发展有什么期待？

**唐刈：**之前跟 Rust 团队提到过让 Rust 能

跟 C++ 很好的整合，但现在看起来难度还是很大，我们还是只能通过 C。

对于 Rust 的未来，我们希望能提供更加完善的 profile 工具，譬如 memory profile 这些。

再就是希望在网络编程方面能够更方便些，这样大家就能更好的用 Rust 进行高性能服务器开发了。

另外，希望能有更多的人参与到 Rust 社区里面，一起把这门语言发展壮大起来。

### QCon：Rust 学习曲线比较陡峭，可以分享下学习 Rust 的经验吗？

**唐刈：**对于 Rust 学习，我觉得还是要动手实践。就看看语法，看看书啥的不管用，实际中一写代码就编译报错了。大家可以参与一些知名的开源 Rust 项目，譬如我们的 TiKV, rust gRPC 这些，也可以自己写一些小的 library 放到 crate 上面供大家使用。

在我们团队里面，大家也并不是立刻能快速上手 Rust 的，毕竟这门语言学习曲线太陡峭，大家也都是一边开发，一边学习，通常都经过了 1 个月的磨合期，才慢慢熟练的。

**唐刈**, PingCAP 首席架构师，典型技术极客一枚，在分布式，高性能，高可用上面有丰富的开发经验，现正从事下一代分布式数据库 TiDB 以及分布式存储 TiKV 的开发，致力于在基础架构数据库领域，提供一套完备的 HTAP 解决方案，解放生产力。开源爱好者，知名开源软件 LedisDB, go-mysql 等系统的作者。工作之余，喜欢阅读和写作，希望自己不断精进。

# 落地机器学习前，我们应该思考清楚的几个问题

作者 郭蕾



随着大数据技术的普及和发展，机器学习技术从搜索引擎开始，伴随着互联网、移动互联和共享经济的发展，日益成为改善用户体验、提高产品质量的核心组件。机器学习技术不仅给互联网带来了源源不断的收入（互联网广告），甚至也在改善每一位用户的日常电商购物、交通出行、旅游等方方面面的体验。那如何在已有的系统或者业务中落地机器学习相关实践呢？在开始使用机器学习之前，企业应该有哪些准备？带着这些问题，InfoQ 记者采访了 PayPal 大数据研发架构师张彭善。

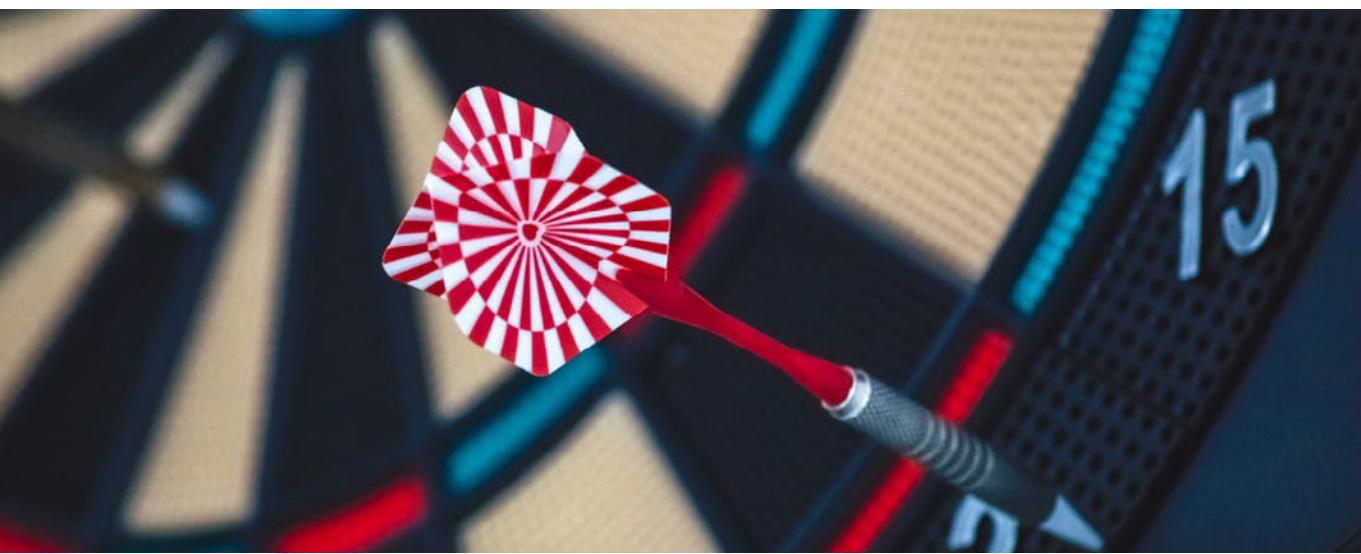
InfoQ：机器学习现在非常火热，很多企业也逐步开始考虑在业务中实践机器学习，你认为机器学习适合解决什么样的问题？

张彭善：抛开 AI 不谈，机器学习在很多领域已经证明了其具有非常巨大的价值，尤其是在互联网领域，无论是杀手级应用互联网广告、还是电商内容平台推荐、风控欺诈识别以及共享出行，机器学习已经成为互联网变现提高盈利的重要手段。从这些主流应用中可以看出目前的机器

学习主要是在解决复杂的预测优化问题，所以对于企业来讲，应当思考业务环节中任何涉及优化策略的业务都可以尝试实践机器学习，比如新产品目标用户推广、销售策略优化、客户流失预警等等。

**InfoQ：那在落地机器学习之前，企业应该有哪些准备？什么样的团队适合实践机器学习？**

张彭善：我和很多企业聊过，这个问题和第一个问题构成目前企业应用机器学习最大的痛



点。对于大部分企业来讲不会像互联网企业一样有强大的数据科学和工程团队，但是我也注意到很多传统企业也是有数据分析和业务开发团队的，这是一个很好的基础，利用这些团队可以组建或培训适应机器学习应用的数据科学团队。然后可以着手先尝试以最小成本构建起端到端的机器学习管道，这包括基础平台的构建、收集数据的渠道、特征工程和模型开发的流程以及最终模型的部署上线，有了这样的管道和平台，就可以持续不断的优化性能进而达到预期的业务目标，然后可以慢慢尝试应用到其它业务场景。

**InfoQ：结合你的工作经验以及业界的一些实践，你认为企业落地机器学习的最佳路径是什么？**

**张彭善：**始终以业务优化作为指导，无论是任何一个步骤，都需要思考是否是有利于达到最终的优化目标。最佳路径是在构建整个流程管道和平台的同时，力争快速部署模型到系统中去。当模型部署上去之后，通过反馈结果再持续的迭代和改进机器学习平台和整个管道，此过程中没

有必要去追求完美的模型和完美的系统，快速响应业务和持续演进能力是关键。

**InfoQ：在落地机器学习的过程中，企业应该注意避开哪些坑？**

**张彭善：**落地机器学习的过程，确实会有很多的坑，比如：

- 学习目标应尽可能的简单以及可以量化，如果目标定义模棱两可，很难想像能做出很好性能的模型。
- 应当首先从简单线性模型开始，不要上来就盲目迷信复杂的非线性和深度模型，事实上每一个模型都有其适用场景和优缺点，从线性模型出发首先可以提供一个好的基准，其次也能帮助发现数据中存在的问题，比如说剔除一些噪音数据，解释一些复杂的特征数据等等。
- 确保训练数据、测试数据和线上数据的分布一致，这个说起来简单，实际上随着业务的复杂性增加，很多情况下训练数据和线上数据并不一致。有可能是业务调整导致线上预



测只是之前的一小部分数据集，也有可能训练数据集只是取的业务其中有偏差的一部分数据。

- 始终以机器学习平台管道构建为目标，不要局限于短期的性能改善，作为一个优化系统，更多的尝试意味着更大可能的性能提高，有了平台的基础，才能持续稳定的改善机器学习模型的性能，提高业务收益。

#### **InfoQ：在一个相对完整的机器学习平台中，你认为应该包含哪些技术栈？**

**张彭善：**机器学习平台是一个相对复杂的系统，主要包括离线、近线、在线三个方向的技术栈。

1. 离线系统的技术栈主要是构建在Hadoop系统，如日志数据存储（HDFS）、特征数据仓库（Hive, SparkSQL, …）、ETL计算、MapReduce特征工程、模型训练和评估，目前一些主流的基于GPU的深度学习也可以统一到YARN、Kubernetes等系统中进行统一的调度管理。
2. 近线主要涉及到一些流式的计算，如计算各

个产品维度的滚动值、历史值等，涉及到的技术栈包括消息系统如Kafka、流失计算如Storm、Flink、Spark Streaming等。

3. 在线主要涉及到在线特征管理和计算以及模型执行，在线系统对于系统的可用性有很高的要求，主要涉及的技术栈包括各种NoSQL数据库用来管理和快速存取特征，Rest或微服务架构管理各种模型和所依赖的特征。除模型预测外，在线系统还可能包括策略系统：主要根据模型预测做相应的策略，此系统一般会和业务系统结合也是偏向服务化的技术栈。

除此之外，还会有监控、日志收集、调度等系统辅助整个平台的运行，这些技术栈并不是特别固定，可以根据实际需要选取和研发。

#### **InfoQ：做应用的时候，大家都经常喜欢说架构。机器学习平台有架构之类的说法吗？**

**张彭善：**机器学习平台并没有脱离软件工程，架构在整个平台依然扮演着非常重要的作用。

- 和传统应用一致，在线机器学习平台目前主

要也是分层的微服务架构，通过微服务可以把数据存取及模型执行以及策略执行做很好的解耦，数据服务通过数据库存取一般是IO密集型的，模型和特征计算是CPU密集型，各服务和应用可以单独优化和部署，进而提高整个在线系统的性能和可用性。如果在线系统涉及到模型增量运算，对于在线系统还需要做好线上模型更新和隔离，这时分层和服务架构更加重要而且对层次划分也更细致。

- 机器学习平台还涉及到很重要的离线数据平台，底层一般是构建在Hadoop/YARN之上的资源管理系统，数据和特征存储在上层的HDFS/Hive或其它Hadoop数据仓库，此外构建基于内存的分布式迭代计算框架，也可以使用Spark或者TensorFlow等主流框架，再上层构建完整的机器学习特征工程和模型训练的管道。

**InfoQ：你认为一个机器学习从业人员和传统应用工程师（程序员）最大的区别是什么？他们的能力侧重点各是什么？**

**张彭善：**从机器学习的应用来看对于传统的应用工程师也提出了新的挑战，首先机器学习应

用是一个融合了离线平台和在线平台的更加复杂的应用，机器学习从业者主要包括数据科学家和机器学习工程师，他们往往是围绕数据和优化目标来设计系统，而为了很好的达到优化目标和业务预期，实际机器学习从业者需要对业务、统计和机器学习以及工程领域都要有很好的理解，而他们的相对弱势是对于系统架构如可用性和服务化等的掌握。传统的应用工程师在这方面较有优势，工程能力是一个很好的基础，也可以帮助他们扩展到面向数据和业务的机器学习应用系统。

**InfoQ：在QCon上海全球软件开发大会上，你会重点为参会者分享哪些技术点？**

**张彭善：**这次也是我第一次作为出品人参加QCon，出品专题是《机器学习应用与实践》，主要是由来自携程、PayPal、美团点评以及唯品金融的机器学习专家分享他们在各自公司运用机器学习技术提升业务价值的实践和经验。我觉得这些议题具有相当的代表性，各位专家也是从另一个角度来帮助我回答上面的一些问题。从应用出发，使用机器学习去驱动业务，不是为了用而用，也是出品这个专题的初衷。

**张彭善，** 2008 年硕士毕业于上海交通大学，2012 年初加入 PayPal Risk Data Science 团队。2013 年初开始研发基于 Hadoop/YARN 的机器学习框架，以满足 PayPal 日益增长的风控大数据的需要。主要负责使用 Hadoop/YARN 实现分布式的神经网络、逻辑回归以及梯度提升树等算法。目前在 PayPal Risk 负责分布式机器学习的框架的研发以及机器学习工程化的端到端的系统平台建设。个人的兴趣主要集中在分布式的计算系统和大规模的机器学习 / 深度学习方向。

# 如何理解Serverless?

作者 郭蕾



Serverless 概念的诞生由来已久，但至今仍然没有清楚的定义。这个词首次出现大概是在 2012 年，云基础设施服务提供商 Iron.io 的副总裁 Ken Fromm 在一篇文章中阐述了为什么他认为 Serverless 是软件的未来。紧接着，在 2014 年，亚马逊发布 AWS Lambda，在这之后，Serverless 开始变得流行起来，国内外各大云厂商都争相跟进。为了进一步了解 Serverless 的概念，InfoQ 记者对亚马逊中国研发中心首席架构师蔡超进行了采访。

Serverless 概念的诞生由来已久，但至今仍然没有清楚的定义。这个词首次出现大概是在 2012 年，云基础设施服务提供商 Iron. io 的副总裁 Ken Fromm 在一篇文章中阐述了为什么他认为 Serverless 是软件的未来（事实上，Iron. io 自 2012 年以来就支持这种概念和框架）。紧接着，在 2014 年，亚马逊发布 AWS Lambda，在这之后，Serverless 开始变得流行起来，国内外各大云厂商都争相跟进。

2016 年 8 月，martinfowler.com 上的一篇文章对 Serverless 做了详细阐述。简单来说，Serverless 主要用来描述两个互相有重叠的概念：

- Serverless 最早用于描述那些大部分或者完全依赖于第三方（云端）应用或服务来管理服务器端逻辑和状态的应用，这些应用通常是富客户端应用（单页应用或者移动端 App），他们建立在云端服务生态之

- 上，包括数据库（Parse、Firebase）、账号系统（Auth0、AWS Cognito）等。这些服务最早被称为“（Mobile）Backend as a Service”，下文将对此简称为“BaaS”。
- Serverless 还可以指代这种情况：应用的一部分服务端逻辑依然由开发者完成，但是不像传统架构那样运行在一个无状态的计算容器中，而是由事件驱动、短时执行（甚至只有一次调用）、完全由第三方管理。对此有一个叫法是 Functions as a service（FaaS）。AWS Lambda 是目前的热门 FaaS 实现之一。

为了进一步了解 Serverless 的概念，InfoQ 记者对亚马逊中国研发中心首席架构师蔡超进行了采访，同时，蔡超也将会在 10 月 17 日举行的 QCon 全球软件开发大会上分享相关话题，欢迎关注。

### InfoQ：可否用通俗易懂的语言谈谈什么是 Serverless？

蔡超：Serverless 还处在一个比较初期的阶段，目前也没有权威和官方的定义。顾名思义，Serverless 就是指应用的开发不再需要考虑服务器这样的硬件基础设施，基于 Serverless 架构的应用主要依赖于第三提供的后端服务（BaaS，Backend as a Service）和应用逻辑运行容器（FaaS Container，Function as a Service）。

打个形象的比方，假如你想卖一些自己的产品，那你就需要买或者租一个店面。但无论是买的还是租的，你都要做一些基本店面布置和维护工作，比如布置柜台、清扫店面，这就好比传统的应用架构，你需要设计，部署和维护硬件基础设施。当然现在你也有另外一种选择，就是在“疯果盒子”这样的店面里，直接租一个放你的产品

的盒子，委托“疯果盒子”来销售你的产品。这就像 Serverless，你利用第三方的服务和设施来实现你的业务。

### InfoQ：Serverless 的诞生到底是想解决什么问题？

蔡超：我想这个问题可以从 Serverless 的主要优势来看：

1. 降低硬件基础设施的部署和维护成本。
2. 降低应用扩展（scaling）成本。目前的 Serverless 运行平台如 Amazon AWS，都可以支持动态的自动扩张。

这些无疑都有助于让开发者专注自身应用的开发，更快地发布他们的应用。Serverless 可以让开发者不必担心基础架构的伸缩性（scalability）是否能支撑未来的业务扩张，同时减少由于访问量变化而带来的硬件资源浪费。

### InfoQ：Serverless 和微服务之间是什么关系？

蔡超：微服务和 Serverless 架构都实现了细粒度的部署，维护和伸缩（scaling）。目前的 Serverless 实现通常会依赖 FaaS，那么粒度就由 Service 变成了更细粒度的 Function。

### InfoQ：Serverless 架构与 FaaS 概念是一回事吗？FaaS 与 PaaS 又有什么区别？可以理解 FaaS 是 PaaS 的升级版本吗？

蔡超：Serverless 和 FaaS 本质上不是一回事。例如，一个简单的 Serverless 应用可以仅仅依赖于 BaaS（比如 AWS SQS、DynamoDB）。目前，大家一提到 Serverless 就会自然的想到 FaaS，主要是因为 Serverless 这个词是在 2014 年 AWS Lambda（Amazon 的 FaaS Container）发布以后才开始变得流行开来的。

我认为 FaaS 和 PaaS 关键的区别在于 Scaling, PaaS 无法像 FaaS 那样在很短的时间内启动或停止一个实例，所以无法实现 FaaS 的按请求动态伸缩，用户必须提前考虑 PaaS 基础设施对于访问量变化的影响。我认为 FaaS 不是 PaaS 的升级，目前看各有特点适用于不同的场合。

**InfoQ：要使用 Serverless 架构，是不是就意味着必须上云？**

**蔡超：**至今 Serverless 没有十分官方和明确的定义。目前的 Serverless 应用都依赖于第三方提供的后端服务（BaaS）和逻辑运行容器（FaaS Container），所以目前的人们认为的 Serverless 的确是运行在云上的。

**InfoQ：我们应该如何构建 Serverless 架**

**构？有哪些需要注意的点？**

**蔡超：**这个问题其实蛮大的，Amazon 给出了一些 Serverless 架构的模版大家可以参考。和其他技术，尤其是新技术一样，在实践中你会遇到不少问题和陷阱，所以值得注意的地方不少，我总结了一些我们的实践经验将在 QCon 演讲中分享给大家。

这里我就提一下最重要的一点，就是首先判断一下 Serverless 架构是否适合于你的应用。例如，如果你的应用是实时性要求很高的，就目前来看是不适合的（关于 AWS BaaS 及 FaaS 的限制你可以参照 AWS 的各服务的官方文档有关于限制的部分），当然你要考虑因素还应包括开发团队和成本等。

**蔡超**，拥有 15 年的软件开发经验，其中 9 年任世界级 IT 公司软件架构师 / 首席软件架构师。2010 年加入 Amazon，曾领导开发了亚马逊全球的新外部直运（External Fulfillment）平台，亚马逊物流 + 系统及基于机器学的亚马逊全球客服系统智能化项目。现做为亚马逊（中国）技术团队的首席架构师，领导中国团队通过机器学习，大数据及云计算新技术不断创新为中国客户打造更好的体验。

在加入亚马逊前，曾任 HP（中国）移动设备管理系统首席软件架构师，该系统曾是全球最大的无线设备管理系统（OMA DM）（客户包括中国移动，中国联通，中国电信等）；曾任北京天融信网络安全技术公司，首席软件架构师，领导开发的网络安全管理系统（TopAnalyzer）至今仍被政府重要部门及军队广为采用，该系统也曾成功应用于 2008 北京奥运，2010 上海世博等重要事件的网络安全防护。

# 道器相融，论一个优秀机器 学习平台的自我修养

作者 蔡芳芳



2017年6月，腾讯正式开源面向机器学习的第三代高性能计算平台Angel，在GitHub上备受关注。作为Angel的主要开发者和团队负责人，同时也是Spark的早期研究者和布道者，他的工作经历可以说同步了通用大数据平台到专用机器学习平台的转变历程。因此，在这之前，InfoQ对黄明进行了一次采访，他将与大家分享人工智能时代的大数据平台演进之路，并结合Angel的开发经验，谈谈如何打造一个优秀的机器学习平台，以及开源后Angel的最新消息和未来规划。

人工智能到底会给企业带来什么？——是改变，机遇，还是更大的挑战？

在之前的大数据时代，企业开始意识到数据的重要性并着手搭建自己的大数据平台，大数据成为业界关注的焦点，Hadoop、Spark等各式各样的大数据框架、组件、平台层出不穷。随着人工智能时代来临，大数据平台发生了新的变化，也提出了更高的要求。Spark等大数据平台多是

为通用数据处理而设计的，并非专用于机器学习任务，企业该如何更好地利用机器学习、深度学习等人工智能技术从实时数据中高效挖掘有价值的信息？

在过去的两年（2015-2017），随着机器学习、特别是深度学习在多个领域取得革命性成功，各种专用的机器学习平台应运而生，百花齐放，Angel也是其中的一员。

# 人工智能时代，大数据平台的演进之路

**InfoQ：您不仅是 Angel 的主要开发者和团队负责人，还是 Spark 的早期研究者和布道者，并且一直从事分布式计算和机器学习领域的开发工作。能否结合您的工作经历，为我们介绍一下通用大数据平台到专用机器学习平台的演进历程？是什么推动了这一转变？您认为将来大数据中心的大多数任务会变成机器学习任务吗？**

**黄明：**其实推动这一转变的，本质上是人们对更高层面的追求所驱动的。从了解过去，到预知未来；从有限空间的穷举，到无限空间的探索；从有监督的训练，到无监督的自我学习……无论是企业高管，还是产品用户，大家都希望能得到更加智能的服务，而也只有提供了这种服务级别的产品和公司，才能在残酷的互联网竞争中胜出。

2010 年，业界的大数据刚刚兴起，当时有很多受欢迎的项目都是统计类的，可以告诉大家昨天最流行的是什么。底层的框架是 Hadoop 和 Hive，很多平台的最大的功能就是出各种各样的报表，天报表、月报表……这时的层次是知道发生了什么。

2012 年，当时有两大发展方向，一种是更快的 SQL，一种是机器学习，涌现了很多开源项目。Spark 能够胜出，是因为它在两者中间取得了均衡，并展现了机器学习的潜质。Matei Zaharia 等人在 NSDI 的 RDD Paper[1] 提到了，Spark 的目标是为了解决 iterative algorithms 和 interactive data mining tools 这两类问题，这个判断从现在来看，依然是正确的。Spark 后来普及开来，目前很多公司依然会把 Spark 当成他们首选的通用数据处理平台兼机器学习平台。

这是人们希望知道即将发生什么。

到了 2014 年，李沐等人在 osdi 关于 Parameter Server 的 Paper[2] 中，给出了分布式机器学习一个更好的思路，后面 Petuum 和 DMLC 的 ps-lite 就出来了。Spark 当时的跟进不是很到位，而且本身 RDD 的理念和 PS 也有些冲突。我们当时还给 Spark 提过一个 PR[3]，后来也没被接受，但是引出了 Glint。到现在为止，官方的 Spark 依然以 RDD 为核心来实现机器学习算法，这是个很大的约束和障碍。

但是在 2015 年，PS 的发展也受到了深度学习的冲击，随着 TensorFlow 的出现，大家纷纷转向了深度学习的框架开发。包括微软的 DMTK 转向 CNTK，DMLC 的 PS-Lite 转向 MXNet……但是实际上，很多公司的数据中心依然有大量 CPU 机器，大量的非深度学习算法还是需要在大规模数据集上进行分布式训练，这个领域是有空缺的，深度学习替代不了。

腾讯是 2015 年开始调研和开发 Angel，其目的就是为了填补上面所说的空缺，2016 年 Angel 开始在内部使用，到了 2017 年，Angel 终于开源，整个开源的过程还是很不容易的（详情可查阅 InfoQ 早前的报道 [4][5]）。希望 Angel 能把这块空白填补上，成为一个专用的分布式机器学习平台，服务更多的公司内外产品，推动人们对更高层次的追求。

最后，未来数据中心，相信依然会有很多数据处理任务。因为无论什么样的模型和算法，其前提都要建立在干净的数据之上。脱离了完整的数据预处理流程，谈机器学习和人工智能都是不现实的。但是大部分数据任务，它们的最终出口和末端，也将会是机器学习和人工智能的任务，因为如果没有了这个终点，前面的数据处理，也

将是毫无意义的，能把控住末端的人，才是最终的胜利者。

**InfoQ：大数据平台早期多以离线批处理为主，实时计算作为补充，但是现在越来越多应用场景对数据处理有了高时效性的要求。腾讯大数据平台也历经了离线计算、实时计算、机器学习三个阶段的发展。未来批处理计算和实时流式计算在企业构建 AI 平台的基础架构中将分别起到什么作用？**

**黄明：**对一个高科技企业来说，实时计算能力与机器学习能力，都是 AI 能力的基础，确实是必备的。而且机器学习的训练（Training）和推理（Inference）两个阶段的重要性会并驾齐驱，实时计算能力的优势还将进一步辐射到推理场景。但是这并不代表离线批量计算不重要了，尤其是训练阶段，离线批量计算依然是主要场景，原因是：

1. 好的模型需要大量的数据，反复迭代并达到一定的精确度才能上线，尤其是效果好的深度学习模型，通常需要多张 GPU 卡，训练较长的时间，才能完成，所以这里高性能的分布式机器学习平台，必不可少。
2. 有很多的算法和场景不支持实时更新，本身就存在约束，或者数学证明不成立，或者不支持流式叠加，所以模型还是需要离线训练好之后，再推送到手机端或者其它终端设备。
3. 在线学习（Online Learning）这个领域，模型的持续优化和更新的非常重要，但是总是需要一个基础模型，而这个基础模型的质量，很重要的制约了后续的改进效果。

综合以上 3 点，离线批量处理，依然会是非常重要和核心的场景，不可替代。但是实时流式

计算会快速发展，尤其是在推理阶段。主要是因为在深度学习时代：

1. 模型比以前复杂，从浅层模型变成了深度模型，其推理计算不是简单的代数计算。
2. 传输数据比之前大，输入很可能是图片、声音、文本等，对吞吐量要求很高，而对推理过程依然要求在毫秒级别完成。这对推理的性能有更高的要求。

所以相信在未来 1-2 年，这方面，从硬件到软件都将会涌现出很多优秀的初创公司。

## 一个优秀的机器学习平台是如何炼成的

**InfoQ：计算是机器学习平台的基础，但不是全部，在您看来，一个优秀的机器学习平台需要具备哪些特性？**

**黄明：**在机器学习界，有些人喜欢把调参和训练的过程，比喻为炼丹，上升到“道”的层面。而道器相融，在我看来，那炼丹就需要一个好的丹炉了，也就是一个优秀的机器学习平台。它需要能为炼丹提供合适的火候，也就是为创新的模型和算法提供最佳的运行环境。因此，一个机器学习平台要取得成功，最好具备如下五个特点：

### 1. 精辟的核心抽象

一个机器学习平台，必须有其灵魂，也就是它的核心抽象。当这个核心抽象和它要面对的模型和算法匹配时，这个平台就成功了一半。如果一开始就错误了，例如 SQL 作为平台的核心抽象，那么对后期的发展制约将会非常明显，无异于缘木求鱼，无论怎么努力都不会成功的。

Spark 的 RDD 核心抽象，很好的解决了分布式大数据的通用问题；而 TensorFlow

中 Tensor、Mutable Variables 和 Dataflow Graphs 的 3 个核心抽象，高度概括了深度学习中的各个元素。Angel 目前的核心抽象是 PSModel，重点解决了分布式机器学习中模型切分，数据并行和模型并行，模式异步 这 3 大问题，基本上可以满足大部分非深度学习的机器学习需求。

## 2. 充分的性能优化

在核心抽象正确的大前提下，性能是决定速度的关键。这就涉及到了平台层对硬件层的理解、调优和封装。去年我们用 500 台高性能机器，获得了 TeraSort 比赛的冠军，也是这种性能优化能力的体现，并将其嫁接到了 Angel 之上。

现在已经不是 MR 的时代走海量低配机器路线。无论是 CPU 机器，还是 GPU 机器，都在往更强更快的方向走。去年比赛我们用的是很高性能的单机，包括 IBM 的 PowerPC，512G 的内存，多个 NVME 的 SSD，RDMA 的 100G 网络……都是业界顶配。

但是光有硬件堆砌是不够的，平台要对硬件有充分的利用。对于非深度学习，Java 系的莫过于 JVM 的调优了。怎样更好地使用内存，避免 FullGC 的产生，尽量让计算不落地，预读数据流水化处理……这些都是对平台设计的考验。而对于深度学习，CUDA 和 OpenCL 的性能利用，显存和内存的数据拷贝，浮点运算和定点运算的选择，一机多卡的内部通讯……平台都需要进行很多调整，甚至于引入像 XLA 这样的黑科技。

既然是分布式机器学习平台，肯定会涉及到分布式的拓扑结构。目前来看，比较成熟的分布式拓扑结构依然是 MR、MPI、PS 这 3 者。机器学习中，基本上 MR 已经出局了，MPI 凭借深度学习卷土重来，和 PS 分庭抗礼，当然也有整体用

PS、局部用 MPI 的做法，这也未尝不可。在确定网络拓扑之后，就要考虑网络加速了。RDMA 和 NVLINK 这 2 个关键技术很值得关注，也是未来的方向。毕竟数据是直接显存落显存，还是走两次内存，差别是可想而知的，再加上不需要 CPU 开销，对性能带来的影响还是很可观的。

所有这些优化，最后暴露给平台用户的，最好是越简单越好，平台能够依据简单的参数，自动选择最佳的性能通道，这才是对算法工程师和数据科学家意义最大的。

## 3. 强大的容错能力

谈到容错，不得不提一下 MPI 和 MR。在 Hadoop 时代，海量低配机器理论的盛行，使 MPI 被 MR 打压得很厉害。但是到了深度学习时代，大家发现这些高配机器和 HPC 也差不了太多，十几万台的机器，可靠性还是很强的，出错的概率很低，相比之下性能更加重要了，所以 MPI 这种模式又活了过来。

都是从整体来看，规模上去之后，在大型的数据中心，高配版本的 GPU 机器和 T 级别的训练数据，对容错性依然需要取得一定的均衡，这种情况下 PS 模式仍是最合适的。整体架构包括网络的通讯性能是最灵活和鲁棒的，可以做的容灾措施很多，代价也小。最终能够达到的效果会远胜于简单的定期 Checkpoint。

## 4. 灵活的接口设计

正如大家所知，2017 年 Python 已经借助人工智能成为了第一编程语言。这在某种程度上，当然归功于 TensorFlow 和 PyTorch 的神助攻，但是这个趋势背后有其必然原因。Python 语言的优势在于语法简单、上手难度低，而且资源丰富，有充实的数据、可视化和机器学习算法库，建立

了非常好的生态环境，同时它又能与 C 无缝结合，借助 py4j 还能和 Java 结合。基于以上原因，Python 能够为后台强劲的平台提供友好的接口层，达到简约而不简单的效果，也就难怪它会奇军突起、一枝独秀了。

但 Python 其实始终只是后台接口的体现，决定整体的，还是后台的接口设计，这时架构师的整体设计能力就非常重要了。核心理念的封装和延伸、多个机器学习概念的整合、系统的分层和解耦、多个子系统的一致性，这些最终都会体现到接口上，从而决定用户基于接口编写算法的难度。

## 5. 完善的周边系统

TensorFlow 开源之初，吸引眼球的工具之一，莫过于它的 TensorBoard，惊艳度超越了同期产品。当时还怀疑它是否会部分开源，不开源这个模块。一个好的机器学习平台还是要在周边系统的完善上多做些功夫，如果用户基于你的平台，可以快速地调试和定位 Bug，将会大大增强他们使用的信心，这会对用户形成很强的吸引力，最终也有助于形成更好的生态。

**InfoQ：在您看来，如何才能高效搭建一个优秀的机器学习平台？**

**黄明：**先讲个大家都知道的小插曲：TensorFlow 的前身是 DistBelief，当时并不太受深度学习界待见，大部分人做深度学习，要么 Caffe，要么 Torch，基本忽略 DistBelief，后来 TensorFlow 推出，就很受欢迎。这里有个时间细节，Hinton 是 2013 年加入 Google 的，而 DistBelief 是 2011 年开始研发的，TensorFlow 是 2015 年发布的，Jeff Dean 由始至终，都在 Google 负责这个项目。作为外人，不太可能知道 Hinton 到底对 TensorFlow 做出了什么样的贡献，

但是效果是很明显的。DistBelief 之前工程性太强，对深度学习的模型本质和算法友好度不足，Hinton 加入后，第二代的 TensorFlow 的水准远远超越第一代的 DistBelief。整个系统的设计，从上层到底层，从名字到周边，都透露着对深度学习工程师的贴心理解。这也是 TensorFlow 成功的原因。

所以要设计和搭建一个优秀的机器学习平台，在我看来：

首先，要搭建一支工程和算法模型能力都很强的团队。整体上这个团队需要很好的互补能力，要有算法工程师，也要有系统架构师，大家互相配合。算法工程师的数学功底和表达能力很重要，而系统架构师的理解能力和快速实现能力也很重要。另外最好能将学术界的创新能力和工程界的落地能力结合，才能使系统创新性和可靠性兼得。腾讯的 Angel 项目从一开始，就是北大的博士生和腾讯的工程师联合主导的项目，虽然远比不上 Hinton 和 Jeff Dean 这样的大神级别，但是模式是类似的，这是非常关键的一个要素。

其次，需要有大数据作为驱动。之前我们研究过 Petuum，发现有些理念很不错，但是稳定性非常差，在大数据量下很难跑通，而且搭建也很难。所以在 Angel 的研发过程中，我们始终坚持以大数据为驱动的原则，各种 Tricks 和设计必须以最终压测通过为原则，并紧密依靠内部业务，通过场景落地来检验效果，以此保障系统的设计合理性和可用性。这点对于大公司来说其实没有太大的难度，只要有正确的理念和合作即可。但是这对于小公司来说则比较困难。所以这也是 BAT 等大企业开源的框架，和实验室或者初创公司出品的框架相比的优势之一。

最后，需要保持很快的演进速度。

# CREATE

TensorFlow 现在经常被批评接口改动太快。其实最近 Angel 的接口改动也很多，而且有些不能向后兼容。这其中原因很简单，一个是因为业界的深度学习发展太快，新算法和模型、技巧层出不穷，作为一个平台必须能快速适应，不进则退。另一个原因是开发的人太多，即便是 Angel 目前 Star 还比较少，但是内部大量的并行开发，很难保证所有的模块都是合理的，定期重构是消除这些不合理的唯一方法。整体来看，只要是合理的重构，能提升性能，就标识着这个项目还在快速的生长期中，不失为一件好事。

**InfoQ：创新工场的王咏刚老师在《为什么 AI 工程师要懂一点架构》中提到，研究不能只懂算法，算法实现不等于问题解决，问题解决不等于现场问题解决，架构知识是工程师进行高效团队协作的共同语言。能不能谈谈您对架构能力的看法？**

**黄明：**王咏刚老师说的要懂“一点”。这个词在我看来代表了两个意思：

1. 确实需要懂，不能什么都不懂。企业里的算

法工程师和数据科学家一定要有动手能力，不能整天只会做研究、写Paper，Matlab和单机版的Python试验一下，自己独占一台GPU机器玩得很开心，模型做完了不会上线，沟通一到工程部分就聊不下去……其实是处于一种很不好的状态。这样的AI工程师，除非某方面特别强或特别突出，否则在企业是很难落地生存的。

2. 不能指望懂太多。毕竟做算法和做工程的思维重点不一样，脑回路也不太一样，方法论也不一样。两方面都精通的人才，有，但是难找。这也是腾讯做Angel的初衷和目的，就是让算法工程师不需要懂太多底层框架优化，也能轻松地写出高效的、能分布式运行的生产代码，把一些通用的体系化的系统和架构细节屏蔽掉，这样才能极大地提高企业的生产力。

目前来看，包括 Spark、TensorFlow 这些比较好的框架，也正是因为它们能够使数据工程师和 AI 工程师，在适当屏蔽掉底层的架构细节后，



依然能够写出高效的算法代码，才取得了成功。

## Angel平台的新变化和展望

**InfoQ：通过您之前的投稿，大家对 Angel 平台开源前所做的一系列重构和升级已经有所了解，开源以来想必又有了不少新变化，能否介绍一下近三个月你们对 Angel 平台又做了哪些优化？**

**黄明：**开源以来，Angel 低调的发布了 2 个小版本：1.1.0 和 1.2.0，主要是加入了新的算法和优化方法，加强了稳定性，细化和完善之前的功能。这 3 个月内的优化，以稳定和性能提升为主。因为 Angel 的定位是工业级可用的平台，所以非常看重大数据量下的稳定性和性能，我们公布的算法都是生产验证过。同时我们对 Spark on Angel 的接口进行了反复的重构，尽可能和 Angel 本身的接口接近一致和复用，这方面的工件到时候会在这次 QCon 大会上重点介绍。

另外根据用户的反馈，Angel 开发团队正在

开发 2 个大功能，尚未发布，包括：

1. Python 接口：接口优化和重构，以提升易用性。因为之前宣传的时候，很多用户的第一个问题，就是有没有 Python 接口……所以我们不得不把这个作为第一优先级来满足。
2. Spark Streaming on Angel：支持在线学习，加入 FTRL 算法。就像之前说的，实时性也是机器学习必不可少的。那 Angel 本身不做实时这块，但是支持 Spark on Angel，那通过 Spark Streaming 来接入实时训练，也是水到渠成的事情，成本也很低，不过对 Angel 的 HA 和 内存管理，需要进一步的优化。

这两个新功能应该在下 2 个版本就能够和大家见面了。至于深度学习的支持，其实也在进行了，但是有些难度，会晚点推出。

**InfoQ：开源后这段时间，Angel 平台的推广情况如何？有没有什么印象特别深刻的问题反馈？**

**黄明：**Angel 开源以来，其实我们并没有太

刻意推广，包括我们在 github 上 Public 的第一天（6月16日）都没有准备做任何 PR，不过由于之前的影响力，最终各大媒体都报道了。但是腾讯 TOSA（开源委员会）最近一年对开源项目的扶持非常大，态度也很 Open，所以我们主要是借着腾讯开源的力量在做这个事情，发了几篇文章。目前整体的 Star 数接近 2.5k，我们比较欣慰的是 Fork 和 Star 数的比例比较高的，看得出很多人还是对项目很有兴趣的。整体上，我们还是按照自己之前定好的节奏，小步快跑地进行新功能和版本的研发。

据了解和接触，目前有部分公司（如小米、新浪微博等）正在试用 Angel，也有了不少贡献者。印象深刻的有几个：

1. 华为的一位工程师，项目刚发布不久就提交了一个比较大的PR，帮忙把Netty版本升级了，非常给力。后来他想把GraphX集成进来，但是我觉得这个方向不太对，就 Reject 掉了，不太好意思。
2. 微软LightBGM的开发者之一提了个Issue，和Angel开发GBDT的同学互动了10个来回左右，详细地讨论了机器学习任务中MPI和PS的网络通讯开销到底谁更小的问题，进行了很有意思的学术互动。
3. 海外的一个用户主动帮忙翻译Angel的文档，之前为了开源，团队花了快1个月的时间边写文档边改Bug，所有文档加起来应该有100篇左右，翻译工作量巨大。但现在基本全部都翻译完了。

这些都让我们体会到了开源的力量和益处，一个平台在开源之后，会受到来自全球的关注，只要你用心经营，并保持良好的功能和性能，能帮助到用户，用户就会主动帮你做很多事情。而

你的视野，也会变得更加的开阔。很多外部用户的需求非常客观到位，正是他们推动着我们往前走。

**InfoQ：开源三个月后再看 Angel，与一众机器学习平台相比（比如 Spark、Petuum、GraphLab、TensorFlow），Angel 的优势是什么？Angel 的什么特性最能吸引机器学习开发者？**

**黄明：**首先目前其实 Petuum、GraphLab 都不开源，没有可比性。Angel 在研发初期借鉴参考过 Petuum 的一些思路，但是后来实验中发现，Petuum 在可靠性和稳定性上都达不到工业可用级别，所以基本上也都推倒重做了。

和 Spark 比的话，目前 Spark 的重心还是在 SparkSQL 上，这从每个版本的 PR 数就可以看出来，MLlib 的比例很小。这在某种程度上也是因为 Spark 的 RDD 本质局限导致的。相比之下，Angel 重点是机器学习算法，而基于 PSModel 的编程模型可以让各种机器学习的优化和 Tricks 都很方便地实现，对于算法工程师非常友好。伴随着 Python 接口的提供，这个优势将会变得更加明显。

TensorFlow 目前在深度学习上的地位还是遥遥领先，从 7w 个 Star 数就可略见一斑。但是在多机多卡的性能上 TensorFlow 的 PS 做得并不好，最近发布的最新版本还在尝试走 MPI 路线，这是业界难题之一。Angel 目前不会独立做一套新的深度学习框架去和 TensorFlow 竞争，而是会发挥自身优势，把 PS-Service 做好做极致，来加速并行训练并形成互补。

关于传统机器学习算法的生命周期问题，我觉得不用太担心。很重要的一点是传统机器学习算法比深度学习更贴近 problem solving 而非模

拟智能。深度网络模拟大脑结构，所以在人类擅长的智能领域优于传统算法，比如视觉听觉，各种对外界信号的理解……但是还有一些非智能领域，人大脑展现出各种认知缺陷（cognitive deficit），比如对模式是随机还是真实的判断，对概率的认知，对风险评估等等。这些方面传统机器学习方法仍然更有效，不需要通过大量的野蛮暴力尝试，就能得到更好的结论。也许以后会有所改变，但目前传统思路的机器学习还是有必要的，在很多场合简单有用，包括腾讯的很多场景对此还是有刚需，所以 Angel 还是要把它经营好。

整体上来看，目前 Angel 是业界，包括国内和国外，比较成熟的开源参数服务器框架，能够在十亿级别的维度（其实百亿也可以，只不过没生产完全验证过，所以不这样宣传）、T 级别大小的样本量规模下，开发和运行通用的机器学习算法的平台。另外值得一提的是，Angel 的算法开发难度也比较低，有一位开源贡献者很轻松地在 Angel 上实现了阿里巴巴用于 CTR 预估的 MLR 算法，并贡献给 Angel 社区，这正是 Angel 团队所期待的。

#### **InfoQ：您认为目前机器学习平台还存在哪些问题和难点？未来改进的重点是什么？**

**黄明：**目前机器学习平台还存在 3 个大问题：算力的水平扩展、模型的高效压缩、快速的推理

**Andymhuang（黄明）**，腾讯 T4 专家，Spark 早期的研究者和布道者，对分布式计算和机器学习，有独到的经验和研究。目前于数据平台部担任海量计算组 Leader，负责构建大规模分布式计算和机器学习平台，助力腾讯各大数据和机器学习业务快速发展。

能力。

机器学习平台最大的难题还是算力。性能再强，接口再好用，底层优化再极致，单机能力终有极限，需要能水平扩展，而深度学习大规模通用的多机多卡的分布式方案，目前依然是个难题，即便 TensorFlow 也没解决得很好。这也是腾讯致力于 Angel 系统的原因，希望无论是 CPU 还是 GPU，我们都能提供高性能分布式机器学习方案。

除此之外，利用庞大的集群规模，长时间训练出来的精细模型，其大小一般比较大，像 Angel 训练出来的模型一般都会上百 G，而深度学习的模型，多数也到 G 级别。这么大的模型，需要压缩之后才能给终端使用，如何在尽量减少精度损失的情况下，最大化的压缩模型，也是平台需要考虑的。

最后一个是快速的推理能力。无论是终端推理（手机端，无人驾驶车……），还是服务端推理（广告，推荐……），要求都是一样的，要尽可能速度快和高吞吐量。这时如何合理地利用和设计流式实时系统，来快速接入数据进行推理，也是平台需要考量的点。

在未来一年里，相信大部分的机器学习框架，包括深度学习框架，都会围绕着上述几个问题重点发力。这也是 Angel 需要面对的挑战和机遇。

# All Around AI

## 助力人工智能落地

由InfoQ中国主办的AiCon将重点关注人工智能的落地实践，与企业一起探寻AI的边界。在AiCon上，你将会看到来自创新工场、京东、360、腾讯、微软、知乎、饿了么、摩拜单车等国内外知名企业的[人工智能落地案例](#)，也能与国内顶尖的人工智能专家探讨相关的技术实践。

2018年1月13-14日 | 北京·国际会议中心

### SPEAKERS

#### - 演讲嘉宾 -

刘海锋

京东商城首席架构师



颜水成

360人工智能研究院院长&首席科学家



尹大朏

摩拜单车首席数据科学家



**6折** 最低价限时购票，立减1440元  
团购更优惠

大会官网：[aicon.geekbang.org](http://aicon.geekbang.org) 购票咨询：18510377288 扫码关注大会官网，获取更多大会信息



# 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），它一直是我们了解业界研发趋势，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。

▼ 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！

▼ 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。

▼ 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。