

架构师 ARCHITECT

跨越2014, 展望2015



解读 | Review

安全 云计算 前端
Docker Android iOS

.....

分析 | Analyse

2014年JavaScript回顾
GitHub中国开发者年度分析报告

InfoQ



特刊

ArchSummit

International Architect Summit

全球架构师峰会 2015

2015.7.17-18 中国·深圳·大梅沙京基海湾大酒店

侧重业务场景，引领技术趋势



HOT

9大热门技术专题

研发体系构建 电商和零售业的转型

在线教育 移动化机会

智能硬件 大数据背后的价值

开源与企业发展 企业云化的痛点与实践

互联网金融

INVITE

邀请制闭门会议

在线教育机遇与挑战

开源与企业发展

企业云化的痛点与实践

互联网基因的金融



议题提交开放，折扣购票启动，详情查阅大会官网

购票热线：010-89880682 会务咨询：arch@cn.infoq.com 大会官网：www.archsummit.com

中国技术圈的春天（代序）

由 InfoQ 主办的“QClub：Docker 专场——聚焦国内 Docker 创业以及企业实践”活动圆满结束，此文是 InfoQ 总编辑崔康在开场中的致辞，分享了 InfoQ 与 Docker 的渊源，以及对中国技术圈的期望。

感谢主持人的介绍。作为大会的主办方，我想首先非常感谢今天所有到场的参会者和在看网络直播的朋友，听负责报名的同事讲，大家的参与热情很高，这次大会的报名人数超过了 1 千了，但是因为会场容量所限，今天在现场的大约有 4、5 百位，对于咱们技术人来说，平时工作很忙，加个班也是常事，好不容易歇个周末，还有这么早起来参加我们的会议，还有外地的朋友应该比大家平时上班的时间还早，说明大家对技术、对 Docker 是真爱。

其次，我要感谢一下负责这场大会组织和策划活动的所有同事，昨天我们另外一场活动结束之后，很多同学也没有来得及吃晚饭就赶到这里做准备工作，忙到很晚，我今天早上 7 点半来到会场的时候，大家已经各就各位了，非常辛苦，在这里我也谢谢所有可爱的同事。

其实今天是个非常有意义的日子。就在昨天，InfoQ 中文站刚刚过完了 8 岁生日，我们在中关村创业大街举办了一个小型的庆祝活动。2007 年的 3 月 28 日，InfoQ 中文站正式上线运营，从此中国的 IT 技术人有了一个崭新的学习和成长的平台，InfoQ 中文站从一个不知名的中文站逐渐成长为输入全方位优质内容、对技术人有着深刻影响的媒体。从今天起，InfoQ 中国迈进了第九个年头，就在开启新航程的第一天，我们举办了 Docker 大会，这不是巧合，可以看出我们对 Docker 这个领域的重视程度。

由此我想到了 InfoQ 和 Docker 之间的渊源。我记得在去年下半年的某一天，我们编辑部的郭蕾同学找到我说，现在有个新的容器技术，叫 Docker，挺热门的，想重点做一下。我当时对 Docker 基本没听过，就听他开始讲。很快，我就觉得这是个好东西，因为我是技术人出身，在转型做媒体之前，最后一份技术工作正好是做云平台，而且负责的团队就是做开发和运维，所以大家可以理解，我会多么喜欢这个新的 Docker 技术。所以我非常支持编辑们在 Docker 方面投入比较多的精力来做些事情。当然我低估了郭蕾同学的热情和执行能力，在不到半年的时间里，他策划了 3 个 Docker 专栏、一个 Docker 周报，还有一堆堆的新闻，有段时间，读者会发现 InfoQ 的首页上每天都会有三分之一甚至是一半的内容与 Docker 相关的，后来连我都看不下去了，要求编辑们压缩一下 Docker 内容的比例，注意内容的多样性，避免成为一个 Docker 专业网站。正因为 InfoQ 在 Docker 方面有了很多的知识积累和专家资源，所以才有了今天的 Docker 大会，可以告诉大家，这场大会的内容策划我们只花了一天的时间，包括选题、讲师人选等等，如果没有过去的积累，我们是很难做到的。

最近，业界有个新闻非常火。那就是 Facebook 第一位华人员工赵海平回国加入阿里。相信大家已经都知道了。我们的编辑上周也专程去杭州进行了专访，马上会有两篇稿件出来。另外，4月底的 QCon 北京大会也邀请了他做主题演讲。这样的故事在最近一段时间一直在发生，比如去年 Google 大脑的创始人吴恩达加入百度。大家可以看到，这是一个很好的发展趋势，那就是中国技术圈越来越吸引人才，技术人越来越关注中国 IT 的发展，并且开始引领技术的潮流。我想这有几个方面的原因：

1. 历史的原因，由于 IT 是个相对新兴的行业，特别是在一些垂直细分的领域，中国的技术人和国外大体是在同一个起跑线上，在知识积累和人才储备方面没有传统行业差距那么大，中国没有历史包袱，所以顶尖的技术人能够在中国的技术圈里做出影响世界的成果。
2. 环境的原因，过去 10 年，中国的 IT 行业发展很快，由此我们的技术人各方面有了显著的提升，除了技能和经验方面，还有软硬件的保障，技术人的软硬件走了保障，有越来越多的技术人在物质方面有了保障之后，开始追求兴趣，开始有了更高的理想，想做一些更有价值的事情，而这种基于兴趣和理想的技术驱动力往往很产生惊人的力量。
3. 第 3 点，也是最重要的一点，那就是中国的巨大市场，人口多，消费能力强，潜力大，随便做点什么，就要面临海量数据的处理挑战，而且现在中国的一些技术挑战是世界级的，国外可能没有这样的先例和经验。这些难题对于技术人来说的吸引力是惊人的，也能获得巨大的成就感。所以中国的技术圈越来越有吸引力，越来越有影响力。

回到 Docker 这个领域，现在在中国发展得很快，据我所知已经有五六个 commiter，还有很多的 Docker 创业公司，整体的氛围已经起来了，希望中国的技术圈能够走在 Docker 发展的最前面，成为规则的制定者而不是追随者。而 InfoQ 是技术人的坚定支持者。

今天的会议，我们请到了外企的 Docker 的专家、Docker 的 Commiter、Docker 的创业者，还有国内外企业的专家，大家都属于中国的技术圈，希望大家一起为 Docker 在中国的发展贡献力量。最后，我想引用我们的掌门人，InfoQ 中国的创始人霍泰稳在昨天庆祝活动中分享的一句话来结束今天的致辞，“从来不要怀疑世界是由少数有梦想和坚持精神的人改变的，而事实也是如此”。我相信今天在场的各位就是能够改变世界的少数人。谢谢大家！

——InfoQ 中国总编辑 崔康

目录

解读 2014 之云计算篇

云计算篇：有一种态度叫做“拥抱”（上）

云计算篇：有一种态度叫做“拥抱”（下）

Docker 篇：才气、勇气、运气

解读 2014 之移动开发篇

iOS 篇：拥抱变化

Android 篇：连接世界

解读 2014 之安全篇

史诗级漏洞频发

解读 2014 之前端篇

任性的颠覆式改变

2014 年 JavaScript 回顾

解读 2014 之开源篇

GitHub 中国开发者年度分析报告

解读 2014 之云计算篇：有一种态度叫做“拥抱”（上）

作者 崔康

编者按

2014 年，整个 IT 领域发生了许多深刻而又复杂的变化，InfoQ 策划了“[解读 2014](#)”年终技术盘点系列文章，希望能够给读者清晰地梳理出技术领域在这一年的发展变化，回顾过去，继续前行。

本文为“解读 2014 之云计算篇”，今年，云计算已经不再像前几年那样火热，产业界对云计算的关注度已经被大数据、物联网等新的名词所超越，但这并不意味着云计算本身影响力的削弱，而是因为“云”已经成为 IT 技术和服务领域的“常态”。产业界对待云计算不再是抱着疑虑和试探的态度，而是越来越务实地接纳它、拥抱它，不断去挖掘云计算中蕴藏的巨大价值。

本文试图通过几个关键字来勾勒出整个云计算领域今年发展的脉络，深入分析这些词背后的意义、InfoQ 对此的报道、对势态的评论等。

生态

“生态”是 2014 年云计算领域说得最多的一个词，各大厂商都逐渐认识到云服务不是一个孤立的产品，传统的软件销售模式已经不再适应互联网的要求。客户购买云服务，不是使用它的某种“功能”，而是基于厂商的平台搭建一个系统，规模可大可小，这种服务使用模式要求客户具备全面的技能，面临的困难也接踵而来。基于此，打造一个完备的生态圈，成为云服务厂商的当务之急。

InfoQ 此前[报道](#)过，阿里云宣布启动“云合计划”，基于开放的 API 以及共享的市场资源，对外招募合作伙伴。目标是在未来三年内招募万家以上的合作伙伴，无论是咨询公司、系统集成商、独立软件商、中小服务商、软件外包与 IT 服务商还是数据中心服务商。事实上，据我和多个云服务厂商沟通了解，包括 Microsoft Azure、UCloud 等都推出了相应的创业者扶植计划，不一而足。

既然是生态系统，那么一定要制定并维持好生态系统中的游戏规则。正如我在某云厂商的恳谈会中所说，这个系统不单单是开发者，还需要包括他背后的就职公司、他开发的产品面向的客户、开发者的合作伙伴等等，当然还有云服务厂商。一个平台想要凝聚力，不外乎从物质和精神两个方面分析。物质方面，开发者能够从这个平台上获取利益，比如这个平台是支持开发者创

业的、多语言跨平台的、具备快速部署优势的、适合迭代开发的，生态系统的成功案例多，具备这些特质，会让这个平台的开发者最终受益。从精神方面，要让开发者有参与感，他在开发方面的经验和知识能够通过生态系统分享出去，影响其他人和公司，并且他的意见能够得到及时的反馈，那么这种参与感会吸引开发者。

云厂商的成败关键在于生态圈建设，谁的圈子越大越成熟，谁就越容易活下来。可以预见，在未来两年，厂商的生态圈建设将全面铺开，除了目前已有的创业者支持计划等，更重要的是制定生态圈的生存法则，塑造合作共赢的业务模式，让生态圈里的大大小小的利益相关者都能找到适合自己的定位，并最终盈利。

落地

“落地”这个词在今年更多地用在了全球的云服务提供商。从没有人怀疑中国市场的巨大潜力和利益，云厂商也不例外。从世界范围来看，美国领跑全球云服务市场，云计算产业体系完整，巨头企业加速向全球扩张，据业内数据，目前在全球 TOP 100 的云计算企业中，美国占 84 家。亚马逊占全球 IaaS 市场的 40%、微软占全球 PaaS 市场的 64%，Salesforce 占全球 SaaS 市场的 21%。众多厂商在领跑之后，都把下一个掘金的目标对准了中国，国内的 IT 产业近几年迅猛发展，云计算市场发展势头很好，特别是国内云厂商正在抢占市场份额，让美国公司的落地步伐迈得更紧。

据 InfoQ [报道](#)，微软中国在年初宣布由世纪互联运营的 Microsoft Azure 在中国正式商用，并在年末的时候开放注册，虽然中国版和国外版在某些最新功能上有所延迟，但同步的节奏正在加快。Satya Nadella 在今年 2 月上任微软 CEO 后提出“云优先、移动优先”的口号，Scott Guthrie 则接任 Satya 此前担任的 Cloud and Enterprise 执行副总裁一职。年末的时候，我[采访](#)了 Azure 落地中国的总导演、微软大中华区副总裁兼市场营销及运营总经理严治庆。他指出，作为第一个吃螃蟹的人，国外云厂商需要通过独特的商务和运营模式在中国落地，这既需要和各级政府部门的积极努力，也需要合作伙伴的紧密合作。

除了已经落地的 Azure，还有一个即将落地的大鳄——AWS，作为云服务的开创者，AWS 在国内一直保持的很高的品牌知名度。据我了解，很多有国外业务项目的本土公司已经大量使用了国外的 AWS 服务。面对国内机遇，AWS 着手布局。目前在国内推出了有限预览版，开设了多个数据中心，并在今年推出了中文支持和运营团队，部分企业开始在国内区域上线业务，何时才能真正落地，我们在 2015 年值得期待。

未来 IaaS 和 PaaS 平台的竞争将更加激烈，特别是在国外云厂商进入中国以后。国外公有云进入中国的门槛比较高，受到一些条件的制约。但是中国的

市场潜力巨大，包括亚马逊、微软等 IT 巨头都在进入中国。在未来两年，更多的国际云厂商通过与国内公司合营的方式进入中国，其相对成熟稳定的云服务和较高的品牌效应将抢占国内份额。不过，和众多国外产品进入中国面临的问题一样，如何“接地气”是一个巨大的挑战。我们看到，对于耕耘中国多年的国际巨头，在这方面已经很有经验，构建生态圈，同时给予中国的研发团队足够的话语权，问题正在得到解决。

国外云服务落地中国，一方面是带来竞争，另一方面是繁荣市场。我们应该看到，国内的云计算市场正在处于“落地”的阶段，需要有更多的玩家进来，不论是国外国外的、大大小小的，一起把这个市场做大，让企业更加了解云计算，拥抱云计算，这样大家才有肉吃。如果只是一两家独大，活力不足，整个云计算产业也带动不起来。

收购

“收购”正在逐渐成为各大公司实现战略布局、抢占先机的快捷手段。我们很久就了解到，制约经济增长与经济活动的，并不是资金的供给不足，而是对资金的需求不大。所以，对很多公司来说，担心的不是没有钱，而是有了钱，应该如何进行有效的利用。在日新月异的互联网时代，一方面云计算产业机会很多，给中小型创业公司提供了发展和崛起的空间；另一方面，大厂商应对变化的动作受到机制的约束，缓慢而繁琐。收购成为大公司占坑的好办法。我们来看看本年度比较有名的云计算领域收购新闻：

- 惠普收购云计算软件公司 Eucalyptus
- Docker 收购持续集成服务提供商 Koality
- 爱立信收购云计算软件公司 Apcera 控股权
- EMC 收购云计算创业公司 Cloudscaling
- SAP 收购云计算软件公司 Concur
- 苹果收购云计算公司 Union
- 云计算企业 CallidusCloud 收购 LeadFormix
- 思科收购私人控股 OpenStack 供应商 Metacloud
- 世纪互联收购第一线集团旗下的 DermotHoldingsLimited
- Google 收购云计算工具开发商 Firebase

其中，惠普收购 Eucalyptus 的新闻，InfoQ 曾经[报道](#)过，收购之后，原 Eucalyptus CEO Marten Mickos 将出任惠普 Senior VP、云计算业务部 General Manager，负责 Helion Cloud 的发展并直接汇报给惠普 CEO Meg Whitman。Meg Whitman 表示将投入 10 亿美元打造 HP Helion Cloud 的品牌，而 Helion Cloud 在技术上则跟 OpenStack 项目保持高度一致。

这个名单还可以列很长，不一而足。我们可以看到，收购成为了云计算产业的常态，这也从侧面反映了这个产业正在处于快速发展和变革的阶段。从收购公司的角度分析，收购行为的目的不外乎以下几点：

- 获取人才。有些收购事件，大企业想要的可能不是这些小公司的产品或者核心技术，而是背后的人才我曾经有机会问过多位公司高管：“你们的核心竞争力是什么”，他们的回答通常会归结为一点：人。看起来挺虚的，不过仔细考虑，不论是产品设计、架构设计，还是运营模式、商业模型、资源运作等，都是人在做，通常顶尖的人才比普通人的效率和效益要高几百上千。有时候，少数顶尖的人才做的决策会决定公司的正确走向。所以收购之后，这些人才可能会转而参与更加重要的工作当中。
- 获取核心技术。这在收购事件中比较普遍。一项核心技术的研发有时需要比较长的时间和比较成规模的团队，而且还带着一些运气的成本，再加上存在的专利约束。考虑到这些综合性的因素，从头研发一项核心技术，还是收购一项核心技术，对于大公司来说，收购的性价比更高。这种方式可以让公司快速突破技术障碍，收购之后，核心技术会马上应用到公司的重要产品中，获得技术优势，提高市场竞争力，从而积极影响公司的发展趋势。
- 获取产品，也就是补充产品线。产品和核心技术还不太一样。有些收购，公司是为了把被收购的产品加入到自己的产品体系中。考虑到云计算产品和服务的多样性，大公司不容易做到一个完整的产品线，对于非核心的业务，通过收购的方式纳入一些辅助的产品，从而帮助自己快速构建一个完善的系统，这种方式值得推荐。当然，整合收购的产品是一项非常具有挑战性的任务，之前也有失败的案例，需要公司投入比较多的人力和物力，才能整合成功。
- 获取客户，也就是获取市场。这在收购事件中也比较普遍。通常在这种情况下，被收购的公司和收购公司存在竞争关系，两家各占有一部分市场份额，但是产品和服务都有差别。大公司收购小公司之后，市场份额产生叠加，提高了市场占有率。通常有两种发展方向，一种是保持现有的状态，两种不同的产品服务于不同的客户；另一种则是以一种产品为主，逐渐替代另一种产品，从而以统一的方式服务于客户。目前来看，收购之后的结局大部分是后一种。不过，这种变化同样具有很大的挑战性，如何实现两种产品的统一化服务是个难题，在实践中有成功也有失败。
- 防御性收购。所谓“敌人的敌人就是朋友”。这种情况下，收购公司对被收购公司的人才、核心技术、产品和客户都不感兴趣，大公司通常不希望竞争对手发展壮大，所以对于对手的潜在收购目标都会出手干预，甚至收购过来，要么好好培育，要么慢慢扼杀，总之不能让对手占到便宜。这种单纯的收购方式不常见，一般都是多种收购目的综合作用的结果。

对于被收购公司来说，大多数情况是个好事情，背靠大树好乘凉，市场竞争的激烈让很多创业中小型公司过早夭折，活下来的公司通常都是被收购，极少数才能走到独立发展甚至上市那一步，在当今的 IT 产业中很少见。对于很多公司来说，他们开始创业的时候，最初的目标也许就是被收购。

回到云计算厂商的收购分析上，刚才提到的几种目的在今年的收购中都出现过，不论是哪种目的，这些收购一定是为公司的战略决策服务的，至于收购的效果，还需要更多时间至少到明年才能看出些眉目，现在下结论就太早了。收购作为大公司的一种标配，为云计算产业的布局起到了重要作用。未来还会有更多的收购出现，直到整个产业进入稳定期，只有几家大玩家的时候。但对于云计算来说，还太早，而且很多细分领域不会出现几家独大的情况。

Docker

“Docker”应该是今年最火爆的技术之一，如果没有听说过，那么你有点 out 了。关于 Docker 在这一年的发展情况，可以单独成一篇稿子了，我们 InfoQ 最年轻的高级编辑、也是长期追踪 Docker 发展的创作者郭蕾同学稍后会发表一篇详尽的 Docker 年终盘点，我就不在这里献丑了。我希望从技术发展史的角度来分析下目前 Docker 所处的发展阶段和历史机遇。

一项技术的成功因素其实和谈恋爱一样：在正确的时间、正确的地点遇到正确的人，也就是古语所说的“天时、地利、人和”。回首整个 IT 发展史，我们很难讲某项技术好还是不好，却可以说这个技术是不是顺应了时代的发展需求。一项技术要想获得较好的发展，通常需要满足以下条件：

- 该技术相比之前的解决方案有创新之处，在本质上比较明显的改善——以 Docker 为例，创新分很多种，发明创造是一种，而把现有技术应用到另一个领域解决问题，也是一种创新，Docker 就是后一种，而且解决问题的效果比较明显。
- 应用该技术的领域进入快速发展阶段或者积重难返、亟需取得突破——随着云计算领域的发展，开发者面临着环境管理复杂、软件配置管理复杂等问题，Docker 提供了一种在安全、可重复的环境中自动部署软件的方式，它的出现为基于云计算平台发布产品提供了崭新的方式，这也得益于云计算的发展需求。
- 该技术门槛不高，社区和开发者乐于学习——对于云计算领域的底层工程师来说，Docker 的学习曲线比较友好，没有高深的数学算法，再加上是开源项目，社区参与度高，学习资料和论坛多，所以关注 Docker 的人基数比较大，群众基础好。
- 有厂商愿意推动该技术的发展——刚才提到，Docker 的出现为基于云计算平台发布产品提供了崭新的方式，这对厂商来说是吸引开发者的亮点之一，所以 Docker 一经推出，就吸引了 IBM、Google、RedHat 等业界知名公司的关注和技术支持。

Docker 满足了上述条件，这也是如今火爆的原因。那么，它现在发展处于哪个阶段呢？我们应该如何把握它的发展脉络，是应该全情投入，还是继续观望。我们在这里做一个简单分析。一项技术的发展周期，通常分为以下阶段：

- 播种期——技术概念刚刚提出，原型正在制作，没有任何评估和分析，只停留在纸上谈兵。市场份额为 0。
- 孕育期——有社区和厂商开始关注该技术，并投入资源研究和推动技术发展，讨论其有效性和可行性，培养生态系统，处于起步阶段。
- 成长期——在技术方面明确之后，应用技术的产品和服务快速发展和推出，整个社区和众多厂商推动该技术发展，并获得客户的广泛认可。
- 成熟期——技术研究、应用、评价、路线图都已经非常明确。社区对技术已经完全了解，更多关注应用，厂商的产品已经成熟，并逐渐发现新的问题。市场份额保持稳定。
- 衰落期——技术逐渐无法满足产业发展的需求，新的技术被提上日程，老技术逐渐收到冷落，市场份额严重下降。

如果按照以上的标准来分析 Docker，我们可以认为，它还处于孕育期末尾、成长期开始阶段，社区和厂商开始关注，并投入了较多资源，在业内也产生了一定的声音，也出现了部分应用场景，但大规模的应用还没有推广开来，需要时间的积累。我曾经和国内多家云厂商的专家聊过，他们都很关注 Docker 技术，认为是云计算的一个助力点，但目前都还没有实际应用。一方面是等待技术和社区更加成熟，另一方面也需要投入资源实现技术转型。

由此可见，Docker 的未来趋势已经很明朗，尽早关注和拥抱 Docker 是正确的选择。按照以前的 IT 发展规律来看，规则的制定者要比追随者更有主动权和竞争优势。所以，目前阶段的 Docker 给了厂商和开发社区留下了“上船”的时间和空间，越往后越被动。

InfoQ 一直尝试站在技术浪潮的前沿，让国内的架构师、开发者了解和借鉴整个技术社区的成果。就在几年前，Node.js 刚刚开始萌芽和孕育时，我就策划了“[深入浅出 Node.js](#)”专栏和迷你书，在国内较早地推动该技术的发展和普及，现在 Node.js 已经进入了成长期，我们也为自己曾经做过的努力感到欣慰。

现在轮到 Docker 了，在同事郭蕾的积极推动下，InfoQ 已经成为国内 Docker 技术传播推广的最前沿阵地，“[深入浅出 Docker](#)”、“Docker 源码分析”、“Docker 周报”等热门栏目不断推出，让国内社区和厂商有一个比较全面深入的了解。前几天，郭蕾告诉我，InfoQ 挖掘的 Docker 作者都已经成为国内各个 Docker 社区和会议的带头人了，这是好事，这是我们的价值。在 2015 年，我们还要做更多。

解读 2014 之云计算篇：有一种态度叫做“拥抱”（下）

作者 崔康

编者按

2014 年，整个 IT 领域发生了许多深刻而又复杂的变化，InfoQ 策划了“[解读 2014](#)”年终技术盘点系列文章，希望能够给读者清晰地梳理出技术领域在这一年的发展变化，回顾过去，继续前行。

本文为“解读 2014 之云计算篇”，今年，云计算已经不再像前几年那样火热，产业界对云计算的关注度已经被大数据、物联网等新的名词所超越，但这并不意味着云计算本身影响力的削弱，而是因为“云”已经成为 IT 技术和服务领域的“常态”。产业界对待云计算不再是抱着疑虑和试探的态度，而是越来越务实地接纳它、拥抱它，不断去挖掘云计算中蕴藏的巨大价值。

在[上篇](#)中，我从生态、落地、收购、Docker 等几个方面做了分析，此文为下篇，我会从其他几个方面来勾勒出整个云计算领域今年发展的脉络，深入分析这些词背后的意义、InfoQ 对此的报道、对势态的评论等。

发布

如果翻看 InfoQ 在 2014 年关于云计算方面的新闻，那么新产品、新服务的发布消息无疑占了绝大多数，这也从侧面反映了过去一年云计算领域的进展。简单回顾一下，InfoQ 就曾经报道过以下事件：

- [MongoDB 进一步进军云平台](#)——在 MongoDB 世界大会上，Mongo 宣布在 Microsoft Azure 和 Google Compute Engine 平台上提供企业级服务。MongoDB 因其简单而深受开发者的欢迎，这个完全托管的 MongoDB 服务可能更广泛地采用它的企业级特性。新服务基于 MongoDB 企业版打造，并利用 MongoDB Management Service(MMS) 提供了即时恢复的特性。
- [UnitedStack 发布 UOS 2.0，统一架构的公有云+托管云](#)——OpenStack 初创企业 UnitedStack 团队于年中宣布 UOS 2.0 正式发布。与之前的 UOS 1.0 不同，本次发布的服务是公有云和托管云这两个产品，而不是之前的 UOS 发行版。UOS 2.0 公有云与托管云对用户开放 Web 控制台、CLI 与 API，所有 API 均是标准 OpenStack API。底层的分布式存储与网络、分布式管理、服务器、网络设备、安全防护不对用户暴露，完全由 UnitedStack 团队进行运维与管理。公有云用户可以直接

在 ustack.com 注册新账号使用，云主机、存储、公网 IP 与带宽、路由器的计费标准可在建立新实例的时候实时预览，按秒级计费。托管云客户需要使用 UnitedStack 提供的硬件规范提供硬件，付费标准为 2000 元每物理节点每月。UOS 2.0 的代码对托管云客户开源。

- [AWS 宣布推出亚马逊极光（Amazon Aurora），一种和 MySQL 兼容的 RDBMS](#)——在 AWS 的 re:Invent 大会上，Amazon（亚马逊）宣布了被称为 Aurora（极光）的全新数据库。Aurora 和已经支持的 MySQL、MS SQL Server、Oracle 和 PostgreSQL 一样，用于支撑亚马逊的 RDS（关系数据库服务）。据 AWS 的宣传，Amazon Aurora 是一种关系数据库引擎，它结合了高端商业数据库的高速、可靠性以及开源数据库的简单性和低成本。它的设计目标是在相同硬件条件下运行时，Amazon Aurora 可以提供五倍于标准 MySQL 的吞吐量。作为可管理的数据库产品，亚马逊 RDS 会处理类似配置、打补丁、备份、恢复、故障检测和维修管理任务。另外，Amazon Aurora 可以通过“只有用到才付费”的定价模式来使用。
- [微软推出 Azure 平台机器学习和实时分析服务](#)——微软宣布推出新的 Azure 平台机器学习功能。现在，Azure Marketplace 拥有了更多以 Web 服务形式提供的机器学习功能。来自微软和第三方提供商的大量服务为终端用户提供了从异常检测到回归模型、二元分类、预测等各种方案。例如，现在的 Web 服务中有一个推荐引擎，用于把产品推荐添加到一个网站或者欺诈检测系统。
- [谷歌计算引擎全面提供 Autoscaler 服务](#)——谷歌云平台 IaaS 组件谷歌计算引擎（GCE）具备了 Autoscaler 功能。Autoscaler 可以动态地扩大或缩小部署规模。它可以根据负载情况增加更多实例（扩大）或者删除现有实例（缩小）。该服务可以根据 CPU 负载、目标利用率或者根据云监控服务定义的指标调整实例数量。此外，Autoscaler 还可以与 HTTP 负载均衡器或者网络负载均衡器搭配使用，在一组同构实例之间均衡的分配流量。
- [AWS 发布 CloudTrail 处理库](#)——Amazon Web Services（AWS）发布了 AWS CloudTrail 处理库(CPL)，这是一个“Java 客户端库，使开发人员可以更简便地构建应用程序，以一种具有容错能力和高扩展性的方式读取和处理 CloudTrail 日志文件”。AWS CloudTrail 记录了一个 AWS 账号所做的所有 API 调用，用于日志记录和审计场景，包括安全分析、更改跟踪、法规遵从性援助和运行故障排除，先前的报道对此有更详细的描述。CloudTrail 是在 re:Invent 2013 大会上推出的，经过 2014 年的扩展，已经支持所有 AWS 域和大部分服务。
- [Canonical 推出 Ubuntu for Cloud 精简版](#)——Ubuntu 是云上最流行的 Linux 发行版，其母公司 Canonical 推出了该操作系统的一个轻量级版本 Snappy Ubuntu Core，该版本针对云进行了优化。Ubuntu Core 是 Ubuntu for Cloud 的一种新的呈现形式，具有事务性更新特性。Ubuntu Core 是一个最小的服务器镜像，具有同现在的 Ubuntu 相同的库，但应用程序通过一种更简单的机制提供。Canonical 将这种现代化的网络

级应用程序称为“snappy”应用。如果需要为运行在容器中的应用程序提供更高的可靠性，那么 snappy 应用和 Ubuntu Core 本身可以实现原子性升级和回滚。

以上只是众多新闻报道的一小部分，我们可以看到，不论是国内厂商，还是国外厂商，都在不断推出新的产品和服务。这可以分为几类：

- 数据管理——客户在云服务上的需求重点依然是数据存储和处理，所以，不论是关系型数据库，还是 NoSQL，还是大数据工具，都在不断地出现在云平台上，接下来会不断地优化服务。而数据管理往往也是考核云计算厂商技术能力的一个关键指标，客户在技术选型时会重点关注这方面的服务全面性和可靠性。
- 操作系统——作为客户各种服务的载体，操作系统是云服务的基础软件，毫无疑问，开放的 Linux 是云计算领域发展的大赢家，成为大多数客户的系统首选，连 Windows 东家的微软推出的 Azure 平台都主动地提供了各种 Linux 选项。在未来几年，Linux 发行版还将进行一场激烈的竞争，围绕着容器技术等方面进行更多创新。
- 处理工具——借用长尾理论，其实客户在面对基本上都差不多的关键性服务产品的时候，考虑的更多的是一些“小功能”的区别，这就考验着云服务厂商的产品完备性和服务链，一方面需要云厂商自己开发和提供各种处理工具和辅助功能，另外一方面云厂商的合作伙伴和第三方开发商也会提供相应的支持。通常生态圈越大的云厂商，第三方的处理工具越多越完备。
- 基础服务——云服务的一个关键优势是基础服务做得好，所以很多基础服务都在不断优化，包括上面所提到的谷歌 GCE 提供 Autoscaler 功能、UnitedStack 改变运营模式从发行版到托管云等，都是基础服务的变更和优化。

在 2015 年，预计将会有更多的创新服务和产品推向市场，一方面是把传统的软件服务迁移到云平台上，另一方面针对云平台的特性，提供特有的服务。

开源

回顾 2014，开源不仅在云计算领域新闻不断，而且在整个 IT 领域都有很大发展，变成了一种时髦的“商业模式”。从 InfoQ 过去一年的报道中，我们可以看到：

- [Joyent 开源云计算技术，欲挑战 OpenStack](#)——Joyent 在巴黎召开的 OpenStack 峰会上宣布将其核心产品云管理平台 SmartDataCenter 和对象存储系统 Manta 开源，遵守 MPL 2.0 开源协议发布。SmartDataCenter 是一个开源的云管理平台，是成千上万的客户在 Joyent 的公共云系统和 Dell 数据中心解决方案上使用的云操作系统，它提供了下一代基于

容器、面向服务架构的多数据库中心管理。其重点是为了简化安装和管理。目前 Joyent 公有云和很多私有云都使用该系统进行管理。

- [开源的云计算开发包 Apache jcclouds 1.8.0 发布](#)——Apache jcclouds 是一个开源的云计算 Java 开发工具包,能够帮助开发者进行云计算应用开发,并可重用已有的 Java 和 Clojure 技能。jcclouds 提供了云计算环境的可移植抽象层以及云规范特性,且支持 Amazon、VMWare、Azure、Rackspace、HP 等云计算平台,它旨在提供一个公共接口以管理众多厂商、提供商、框架及 API(从 IaaS 到 PaaS)中的计算机节点和存储节点。jcclouds 支持全世界 30 个不同的提供商,开发者与运维人员可以通过下游工具如 Apache Whirr 或 Pallet 来使用它,也可以直接通过 API 和 Ant task 将其当作库来使用。
- [VMware 宣布其在推动云计算和开源方面的相关行动和规划](#)——在 2014 年 VMworld 大会上,VMware 宣布了他们将会发布一款 VMware 自己集成的 OpenStack 产品,它使用了 VMware 自身的一些软件组件用于存储、网络和管理。这个产品将会在 2015 年的上半年交付。从 2012 年底开始,VMware 公司就成为 OpenStack 基金会的成员,并一直致力于使 OpenStack 支持 VMware 自己的技术,如 VSphere 和 VMware NSX。目前 VMware 有超过 30 名工程师在 OpenStack 项目上工作。
- [Deis 1.0 发布](#)——Deis 是一款基于 Docker 和 CoreOS 的开源 PaaS 平台,旨在让用户能够容易地在服务器上部署并管理应用程序。它能够部署任意可在 Docker 容器中运行的应用或者服务;在语言和框架层面,除了可以使用 Docker 容器之外,Deis 还为 Ruby、Python、Node.js、Java 以及 Scala 等语言提供了 Heroku 构建包。Deis 能够被部署到所有支持 CoreOS 的系统上。最近,Deis 发布了第一个稳定版本 v1.0.0,该版本包含了稳定的 API,广泛的特性以及固定的组件架构,标志着它已经可以应用于产品环境。
- [PaaS 论剑: Cloud Foundry 对决 OpenShift](#)——时至今日,Red Hat 的 OpenShift 和 Pivotal 的 Cloud Foundry 已成为两大主要开源 PaaS 系统,他们分别提供了三个版本:托管、企业和开源。Cloud Foundry 具有广泛的应用支持和一流的易用性等优势,而 OpenShift 有着管理和自动化上的优势。Cloud Foundry 和 OpenShift 究竟有什么不同呢,Infoworld 测试中心对 Cloud Foundry 和 OpenShift 进行了一次对比测试。

正如我在[分析.NET 开源](#)时所说,开源是大势所趋,就在 5 年前,各大公司和厂商还在讨论是否应该开源,但现在大家讨论的则是如何更好地开源。随着互联网时代的发展和成熟,开源已经成为标配。以前,开源是一种保障软件项目生存的方式,而现在很多开源项目已经成为企业的一种商业和运营策略,生存已经不是问题,如何活的更好才是关键。开源的好处是,吸收整个技术社区的力量促进开源项目的发展和竞争力,第二个好处是,提高开发者的参与感和信任度,有利于吸引开发者,构建良好的生态系统。劣势,对于一些

没有厂商支持的开源项目，路线图不明确，有纷争，存在分裂或者失败的机会。从云计算领域来看，云厂商分为了两大阵营，要么基于开源的 OpenStack 和 Cloud Foundry 等平台，要么采用私有技术比如 AWS 等，目前私有技术阵营占有市场优势，但开源阵营步步紧逼，未来趋势还是拥抱开源，大势所趋。

一些观点

这部分是编辑部团队同学的观点，列出来供大家参考：

- 产品线方面，跑得比较快的几家都在不断加入更多（类 PaaS）服务，如 RDS、Redis、代码托管、部署和生命周期管理等
- 技术层面，云服务的性能、稳定性越来越可以跟物理机相媲美
- 技术线方面，各种技术混战的场面逐渐收缩到比较少的几条技术线，跟上面提到的收购有关系，比如开源的云体系基本上收缩到只剩下 OpenStack 了，已经是所有传统 IT 巨头们（IBM、Oracle、华为、EMC 等）的押注对象
- 基于容器的私有云随着 Docker 的火爆而在（互联网）企业中活跃起来，但是基于容器的公有云还是玩不起来，Docker 公司自己现在也还在摸索此类 web service 的玩法
- 想过自己用开源云体系搭建一套内部私有云的企业有些已经尝试过一轮了，市场上（尤其是美国市场）对外部私有云/托管私有云的尝试正在增加，至少开发/测试环境上外部云已经越来越常见
- AWS Lambda 的发布很有意义，可以说是划时代的，但 AWS 的竞争者们大多不同意这个观点……

下篇结束，感兴趣的读者可以查看[上篇内容](#)。本文由 InfoQ 总编辑崔康执笔，编辑部成员均有贡献。



云计算高可用架构设计与实践

阿里云技术专场

名额有限，报名从速

时间：4月23日（周四）

地点：北京国际会议中心



解读 2014 之 Docker 篇：才气、勇气、运气

作者 郭蕾

编者按

2014 年，整个 IT 领域发生了许多深刻而又复杂的变化，InfoQ 策划了“[解读 2014](#)”年终技术盘点系列文章，希望能够给读者清晰地梳理出技术领域在这一年的发展变化，回顾过去，继续前行。

毫无疑问，[Docker](#) 已经成为 2014 年最热门的技术之一，它被爱好者冠以云计算新星、下一代虚拟机等称号，可见大家对其的期望之高。2014 年，Docker 的发展可谓是一路凯歌，从年初的 B 轮融资到年末的 DockerCon 欧洲大会，Docker 在这一年里顺风顺水，就连微软、谷歌、AWS 这样的巨头也敬它三分。InfoQ 从 2014 年 5 月开始重点关注 Docker，先后策划了[深入浅出](#)、[源码解读](#)、[CoreOS 实战](#)、[Docker 周报](#)等系列专栏，共产出了百余篇新闻与文章，同时还建有两个千人的 QQ 群以供读者学习交流。在这一年里，作为负责 Docker 专栏的技术编辑，我密切关注了 Docker 的每一次融资、收购、合作，也亲身见证了一个技术的发展之路。本文作为 Docker 专栏的年终总结，分析了过去一年 Docker 的发展、模式、应用，并结合自己的理解展望了 2015 年 Docker 的发展以及与它相关的技术变革。

另本文的标题来自热播无胸版电视剧《武媚娘传奇》张公公之口。

发展

2014 年是 Docker 的起步之年，这一年里 Docker 又是融资，又是收购，又是办大会，又是与巨头合作，迅速获得了资本以及云厂商的认可。记得在 6 月的一段时间里，每天早上起来只要一打开朋友圈，就会看到关于 Docker 的爆炸性新闻，想想也是醉了。先撇开 Docker 的优势不谈，我们来回顾下过去一年 Docker 的几个重要的发展节点。

1. **Docker 1.0 发布。** 6 月 10 日，[Docker 团队宣布发布 Docker 1.0 版本](#)，从第一个版本到 1.0 版本的发布，Docker 大约经历了 15 个月的时间，共收到了超过 460 位贡献者的 8741 条改进建议。1.0 版本的发布也就意味着 Docker 可用于生产环境，从发布特性中可以看到，Docker 当时重点在关注稳定性、扩展性、兼容性等方面的问题，并没有提及安全事宜。同时从数据中可以看到社区对 Docker 的贡献。

2. Docker 出售 dotCloud。8月4日，[Docker 宣布出售其平台即业务服务 dotCloud](#)，并称接下来将专注于容器业务。CEO Ben 表示他们已经看到 Docker 的快速增长，希望把所有的精力和资源都集中在这项业务上。从这件事情上可以看出 Docker 公司的决心，把成败都押在 Docker 上，何尝不是一种魄力。当然，那时 Docker 有足够多的用户，他们也很有信心。
3. 微软全面拥抱 Docker。10月，[微软宣布下一个版本的 Windows Server 将支持 Docker](#)，同时还发布了 Windows 10 客户端的技术预览版。Windows Server 主要是面向企业市场，显然，微软已经意识到容器技术的发展趋势以及其对企业的意义。比较有意思的是，微软早在之前就开发了自己的容器技术（Drawbridge），而 Windows Server 却选择支持 Docker，可见 Docker 的势头之猛。
4. AWS 加码押注 Docker。在 AWS re:Invent 2014 大会上，[AWS 推出了高性能容器管理服务 EC2 Container 服务](#)，用户可以在 AWS 上使用容器轻松地运行和管理分布式应用。AWS 是云计算的老大哥，在其 EC2 上早就支持 Docker，这次的容器服务瞄准的是集群服务，也是第一家将 Docker 应用于集群服务的公有云，AWS 果然是 IaaS 的带路人和领头羊。
5. Docker 发布新的跨容器的分布式应用编排服务。12月4日，[Docker 宣布发布跨容器的分布式应用编排服务](#)，编排服务可以帮助开发者创建并管理新一代的可移植的分布式应用程序。之前 Docker 关注的都是单个或者少量的容器，所以我们也经常看到官方用词“Engine”，而编排服务的发布意味着 Docker 正式进军集群服务，彻底平台化。通过三个新的编排服务（[Machine](#)、[Swarm](#)、[Compose](#)），用户可以迅速构建 Docker 集群环境并部署应用，至此，Docker 为 2014 年画上了圆满的句号。

模式

正如热播电视剧《武媚娘传奇》中所说，一个才人的上位需有才气、勇气、运气，Docker 在这一年里能有如此快的发展也与此道理大同小异。才气不说，从 Docker 一出现大家就很认可它带来的颠覆性的优势，现在流行打标签，如果要给 Docker 写标签，我认为应该有几个词：容器、虚拟化、轻量、可移植、分布式。谈到勇气，不得不提 Docker 的创始人 Solomon（dotCloud 的创始人），在 PaaS 市场举步维艰的情况下，他敢于将自己的核心引擎开源，并让团队的核心成员参与开源项目，以及后来直接把 dotCloud 卖掉，把全部精力都投入到 Docker 的开发上，可谓魄力十足。运气又名天时，近几年，DevOps、微服务、云计算等技术理念如日中天，而 Docker 可以全部和这些技术集成，并且都能为之一颤。

从[官方公布的数据](#)来看，截止到 2014 年 11 月，Docker 的贡献者已经超过 700 人，与去年相比增长 52%；Pull Request 数量为 5200 个，增长 37%；GitHub 上相关的项目已达 18000 个，增长 177%；相关的仓库数量已达 65000 个，增长 348%；Docker 的下载量有 6700 万之多，增长 2336%。

一年的时间里，Docker 的生态系统发展迅猛，知名的云计算公司以及软件、操作系统、系统集成厂商、配置管理软件、大数据厂商以及开源软件都在向 Docker 靠拢，不管是在哪个领域，Docker 都在“Doing the old thing the new way”。Docker 相关的生态圈可以参考下图，图片来自 DockerCon 欧洲，需要注意的是图中没有列出国内的云计算公司。



应用

前面提到过，Docker 可以和很多的概念联系到一起，生态圈如此之大，那可以做的事情肯定也很多。但是，现在有多少公司/开发者在使用 Docker 了？这个问题真是得打个大大的问号，从平时的采访以及 QQ 群中读者的问题来看，绝大部分人对 Docker 的使用还停留在观望状态，并且将 Docker 应用于生产环境的公司少之又少。记得刚开始追踪 Docker 的时候，我把英文站的一篇[新闻](#)中描述 Docker 发展的词主观的翻译为了“吹捧”，我认为媒体、社区、厂商对 Docker 炒作的声音远大于需求驱动，看似火热而又令人心动的变革其实并不如外表看起来的那么红火。不过，从一个[技术的发展角度](#)来看，如此情况也在情理之中。

Docker 的应用和它本身的优势密切相关，系统集成厂商[Flux7 曾总结过 Docker 的一些应用场景](#)，虽然 Docker 目前的实际应用人数并不多，但也不乏最佳实

践。开发方面，开发者可以使用 Docker 搭建开发环境，借助 Docker 可移植的特性，开发者可以将自己的环境分享给开发以及相关的测试同事，省去了因为环境搭建而耗费的时间。同时，由于 Docker 可以快速创建并启动一个或多个容器，所以它可以和 Jenkins 一起来进行持续集成（CI），相关的开源项目有 [Drone](#)、[Strider](#)。

很少有人提及 Docker 对 SaaS 带来的影响，从普通用户的角度来看，Docker 可以为他们解决很多苦恼的技术问题。举个例子，之前用户想卖东西，可能就是入驻淘宝这样的平台，一是需要依靠大平台的流量，二是自己根本无法搞定一套电商程序的安装以及维护。有了 Docker 后，用户拿到的就是镜像，通过几个来回的命令就可以完成安装。所以我认为假以时日，类似 Wordpress、Drupal 这样的开源软件都会通过镜像的方式交付给用户，用户也会在选择 SaaS 平台还是自己搭建方面有更多的主动权。同样，得益于 Docker，SaaS 平台也许会向 PaaS 过渡。

PaaS 方面，Docker 已经扎根新一代的 PaaS，如果把 Azure、Heroku 等公有的 PaaS 看为第一代，允许用户自建 PaaS 的 Cloud Foundry 和 OpenShift 就应该是第二代，那以 Docker 为首的平台就应该是第三代，主要代表有 [Deis](#)、[Flynn](#)、[Tsuru](#)。目前第三代 PaaS 也已经成熟，其中 Deis 和 Flynn 都已经发布 1.0 版本，借助 Docker，这些平台可以占用更小的资源。部署方面，新一代 PaaS 平台也可以借助 Docker 实现从开发环境到 PaaS 平台的无缝迁移，可谓从里到外，Docker 都是得力帮手。

IaaS 方面，巨头 AWS 已经推出基于 Docker 的容器服务，老二 Azure 也已经全面支持 Docker，并和官方建立了合作关系。放眼国内，阿里云、腾讯云、UCloud、青云等公司都已经支持 Docker，更进一步的支持看似也没有。同时，新一代的基于 Docker 的 IaaS 也在紧锣旗鼓的设计中（此处省去很多创业公司），Docker 可以提高资源利用率，降低云成本，所以也有很提出了 CaaS（容器即服务）的概念。没有人能够知道在这样的时代 Docker 到底会带来什么样的变革，所以面对如此火热的技术，IaaS 厂商扑上去也容易理解。

2015 年

11 月底，[CoreOS 发布了自己的容器引擎 Rocket](#)，Docker 有了名义上的第一个竞争对手。新的一年，容器之争才刚刚开始，其实微软、谷歌都有自己的容器技术，亚马逊应该也有。如果容器会带来历史性的变革，那容器的战争是早晚的事。CoreOS 发布 Rocket 时称 Docker 已经忘记初心，并且开始冲击生态系统中的其它软件，于是便另起炉灶。现在过去一个多月了，还在 Stage 0 阶段，在社区方面获得的支持远不如 Docker 刚开始的时候，所以我并不看好。单从引擎这一点来看，类似 Rocket 的其它容器引擎并不是没有机会，因为 Docker 也有自己的问题（比如安全），新的一年，更是着重考验它们逼格的时候。

DockerCon 欧洲上，Docker 发布新的跨容器的分布式应用编排服务，旨在简化分布式应用的部署。分布式应用可以保证服务的稳定性和可扩展性，Docker 重新定义了分布式应用之道：我们可以根据需求随意收缩分布式应用节点。之前，Docker 教你怎么玩好一个容器，而现在 Docker 会教你如何玩好成千上万个容器，并在这么多容器上部署你的应用。当你在几百台服务器上部署几万个容器的时候，[Kubernetes](#)、[Mesos](#) 这些调度框架的作用就显现出来了，所以分布式应用相关的开源软件也是新一年的风向标。

回到国内，2015 年 Docker 会逐渐落地，经过去年一年的发展，Docker 已经相对稳定，研究和使用的人越来越多，并且中文资料也开始多了起来。现有的云计算厂商应该会重点关注 Docker，并可能在其底层架构中使用 Docker，与 Docker 相关的 IaaS 和 PaaS 也会迎来一个创业小高潮，相关的开发者服务商也会提供对 Docker 的支持。

也许是看武媚娘看多了，每每去思考 Docker 发展的时候，我都会想 Docker 又何尝不是下一个“武媚娘”了？如果是，那“[武代李唐](#)”的预言会不会再次上演？这一次李唐们会如何做？好戏还在后头。



云计算与大数据应用选型

时间：4月24日(周五)

地点：北京国际会议中心



名额有限，报名从速

解读 2014 之 iOS 篇：拥抱变化

作者 唐巧

编者按

2014 年，整个 IT 领域发生了许多深刻而又复杂的变化，InfoQ 策划了“[解读 2014](#)”年终技术盘点系列文章，希望能够给读者清晰地梳理出技术领域在这一年的发展变化，回顾过去，继续前行。

对于 iOS 技术领域，2014 年是变化巨大一年。在硬件上，由于苹果发布了更大尺寸的 iPhone 6 和 iPhone 6 plus，使得 iOS 设备的屏幕适配工作也多了起来。这还不包括全新的 Apple Watch 带来的开发上的变化。在软件上，Swift 语言的推出惊呆了整个业界，从 Objective-C 切换成另一个全新语言的代价是巨大的，但我们也看到了它长远的好处以及苹果对此的决心。除了软硬件外，编程语言、中国区、开发社区在 2014 年也有着巨大变化，我们试着用几个关键词来勾勒出 iOS 技术领域在 2014 年所经历的变化。

硬件

苹果在 2014 年推出了 iPhone 6 和 iPhone 6 plus，iOS 设备也进入了大屏时代！虽然包括笔者在内的许多朋友都认为：“Plus 实在太大了”，但是从销售量看，iPhone 6 plus 在初期占据了 [6 成](#) 的销售量。我本人在实际使用 iPhone 6 plus 几周后，也很快适应了它的屏幕尺寸，现在反而不习惯使用了两年的 iPhone 4s 了。

在开发上，iPhone 6 的推出使得 iOS 开发者也开始像 Android 开发者那样，为适配工作担忧。与 Android 开发者不同，iOS 开发在过去多年都是用“绝对定位”的方式来放置各种 UI 控件，使用 Autoresizing Mask 就可以轻松搞定 iPhone4 和 iPhone5 屏幕高度的差异问题。但现在不一样了，iOS 开发者需要学习使用 AutoLayout，通过各种 Constraint 来调整 UI 控件，但是很多界面是需要用代码来编写的，所以这方面的开发工作会比较辛苦，好在有 [Masonry](#) 这一类的开源库来辅助我们做界面开发工作。

iPhone 6 plus 引入的特有的 3x 的图片问题使 UI 设计师的工作量也大大的增加了。另外，以前由于 Android 设备其实太过于分裂，无法对每种屏幕分辨率都做到精致地设计，所以设计师大多只是用“凑合用”的思想来实施 Android 应用的适配。这一点从 Android 大屏手机上的应用呈现就能看出来。而现在同样的方式不太可能被严格的苹果以及挑剔的 iPhone 用户所接受，设计师很可能需要为 iPhone 6 plus 做专门的大屏设计，才能获得用户的芳心。

Apple Watch 的推出让适配工作真正地进入了 Hard 模式。优秀的开发者需要在工程中做好代码地复用，才能让底层的逻辑能够同样运行在 Apple Watch 中。不过这方面的工作还只是猜想，我们得等到 Apple Watch 真面世的时候，才能真正了解到相关的开发工作有多么不同。

软件

苹果在 2014 年的全球开发者大会（World Wide Developer Conference，简称 WWDC）上并没有发布任何新的硬件产品，但展示出其在软件上的各种功能改进和整合上的巨大努力。苹果推出了新的 OS X 10.10 操作系统 Yosemite，它吸收了从 iOS 7 开始的扁平化的设计风格，很多改进的细节都可以看到 iOS 系统对于 OS X 的影响。Yosemite 最吸引我的是 handoff 功能。handoff 功能将 Mac 电脑和 iPhone、iPad 的功能进行了深度整合，当你拥有这三个设备时，它们每一个都可以接打电话和收发短信，你在它们任何一个设备上编辑的文档，也可以实时同步到另一个设备上。handoff 并不是苹果的应用独有的功能，苹果将其相关的 API 都开放给了开发者，所以开发者也可以完全开发出支持 handoff 功能的应用。

相对于 Yosemite 带来的整合方面的改进，新版 iOS 8 的改进的重点则是开放。苹果开放了输入法接口，允许第三方提供输入法。苹果也开放了 Touch ID，允许第三方应用获得指纹信息。HomeKit 则为智能硬件开放了不少可以互操作的接口。另外，iOS 8 还新增了 4000 个新的 API，开放了大量底层 API 接口供应用调用。所以对于普通用户而言，iOS 8 新增的可见功能可能是有限的，但是其开放的大量 API 和新 SDK 释放出了巨大的想象空间，相信会产生更多有趣的应用。

对于一个硬件来说，相关的软件是非常重要的，但是苹果却将它的操作系统和 iWork 办公套件完全免费，这显示出苹果对于软件生态圈不同的理解。而微软至今最大的两大收入还是来自它的 Office 和 Windows 系列操作系统。在微软还在发愁他的新版 Windows 10 怎么卖出去时，苹果早已不玩这种 1.0 的商业模式游戏了。

Swift

苹果的 Objective-C 语言发明于与 C++ 同时期的上世纪 80 年代初，虽然苹果对其进行了很多次改进，但这终究像给应用打补丁一样，不是特别舒服。这次 Swift 的推出，苹果试图用全新的编程语言来提高 OS X 和 iOS 程序员的开发效率。单从语言特性上来看，Swift 吸收了众多现代编程语言的优秀特性，例如类型推断（Type inference）、范型（Generic）、闭包（Closure）、命名空间（namespace），元组（tuple）等，整体语法上也更加简洁。我们也了解到 Swift 是苹果从 4 年前就开始筹划中的事情，可见苹果在长远规划上

对于 Swift 语言的重视，相信未来 Swift 会接替更多 Objective-C 原本的地位，所有 iOS 开发者都应该花时间学习这门全新的语言。

苹果这次推出 Swift 也吸引了大量的讨论，其中比较不正确的观点是认为 iOS 开发的门槛因此而降低了。从我和身边的同行对 Swift 的学习和讨论来看，Swift 并不是一门可以简单上手的语言。并且在未来一到两年内，iOS 开发必然会经历同时使用 Objective-C 和 Swift 的过渡阶段。对于 iOS 开发者，除了必须掌握以前的 Objective-C 语言的知识，还需要学习并不简单的 Swift，可见 iOS 开发的门槛不但没有降低，反而提高了不少。

中国

苹果是重视中国的，它在大中华区的销售贡献了约 15% 的全球营收。但是从某些方面看，苹果还是不够重视中国的。很多中国用户抱怨的问题，苹果都一直没有解决。我们来看看苹果对于中国区的用户和开发者，还有哪些需要改进的地方。

刚刚提到的 handoff 功能依赖于苹果的 iCloud，但是苹果的 iCloud 存取速度在中国却是非常糟糕的。笔者试图打开一个通过 iCloud 共享的约为 6M 的视频，但是等了 3 分钟才打开。与此同时，国内的各种网盘服务却能提供非常大的免费容量以及非常快的访问速度。

中国的垃圾短信和骚扰电话一直是非常严重的，而 iMessage 免费发送的特点给了垃圾短信发送提供了很好的平台。苹果到现在都没有推出相关的有效避免垃圾短信的办法，只提供了一个用于反馈垃圾短信的邮箱，整个反馈流程也非常繁琐，需要提供发送者邮件、内容以及相关截图。笔者反馈了数十次之后，也没有得到过任何回应。对于骚扰电话，苹果只提供了事后将该电话号码加入黑名单的功能，这使得用户只能是接了骚扰电话后，才能做相应的手工处理。这些问题从技术上要解决没有任何难度，看看 Android 平台就可以看到这方面的问题被非常完美地解决了。但是苹果却不愿意做相关的努力，这只能说明它还不够重视中国用户。

苹果的 App Store 中国区也是比较混乱的，各种刷榜的应用充斥着排行榜的前几名。苹果在这方面应该是做了很多努力的，但是终究还是做得不够。从很多应用的评论中，我们都能看出刷榜的痕迹。由于这方面直接和收入相关，所以我相信苹果还是会尽力改进的。App Store 还有一个问题是支付不太方便，今年苹果和银联合作，不但推出 1 元的应用，而且允许用户使用银联卡来付费，极大地方便了用户购买应用。

社区、会议和第三方服务

国外的 iOS 开发社区今年依然发展红火，比较引人注目的是 objc.io。objc.io 每一期的质量都非常高，所以由知名开发者王巍（onevcat）组织的翻译团队会将每一期文章都翻译成中文。国内的博客和社区发展相比国外还是比较慢，参与分享 iOS 技术的独立博客相比以往多了很多，但整体质量还有待提高。

移动开发也实实在在地成为了一个重要的技术领域，QCon 和 ArchSummit 每一届都会有专门的移动专题，可见这方面聚焦了开发者大量的关注。2014 年国内外都涌现出了大量服务于移动开发者的第三方服务。这些第三方服务作为生态圈的一个重要组成部分，将会方便开发者聚焦核心功能，而不用担心基础设施的搭建。

展望

展望 2015 年，笔者认为：

- 苹果的 iPhone 将会继续占据高端机市场，但千元左右的 Android 机将会占据其它所有的用户，最终 Android 的用户量将远远大于 iPhone 的用户量，但由于 iPhone 用户的优质属性（付费意愿高），所以各大应用将会在这两大平台都会投入足够的开发力量。如果有人能够比较好的解决代码在 iOS 和 Android 平台的复用性问题，将会很大程度上改变现在的开发模式。
- Swift 语言将仍然处于发展阶段，Objective-C 依旧会是 iOS 程序开发的主要语言。更多的开发者将会使用混合开发的方式，同时使用 Objective-C 和 Swift 来开发应用。
- 国内 iOS 开发社区还会进一步发展，会出现更多的移动开发相关的开源基础设施或第三方服务。
- 苹果很可能会继续不那么重视中国区用户。iCloud 服务依然糟糕，iMessage 垃圾短信依然泛滥，由于害怕“侵犯用户隐私”，骚扰电话还是没有很好的解决办法。但苹果在大中华区的收入会继续增加。
- 由于 Objective-C 和 Swift 相比 Java 语言没有那么普及，Mac 电脑也没有 Windows 电脑普及，这造成 iOS 学习门槛较高。优秀的 iOS 开发者依然是稀缺资源，相比同等的 Android 开发者，iOS 开发者可以获得更多的收入。

解读 2014 之 Android 篇：连接世界

作者 郭亮

编者按

2014 年，整个 IT 领域发生了许多深刻而又复杂的变化，InfoQ 策划了“[解读 2014](#)”年终技术盘点系列文章，希望能够给读者清晰地梳理出技术领域在这一年的发展变化，回顾过去，继续前行。

本文为“解读 2014 之 Android 篇”，Android 从 2008 年发布，到 2014 年末已经 6 岁。经历了前几年的高速发展，Android 已经当之无愧的成为全球用户最多的手机操作系统。2014 年虽然不是 Android 发展最快的一年，却是变化最快、扩张最大的一年。最新版本的 Android 5.0 Lollipop 无论是用户体验还是系统性能都有着颠覆性的改变与提升。Material Design 的出现，更使 Android 设备在体验方面第一次和 iOS 站在了同一个高度。

经历了一年多的开发与测试，谷歌也发布了第一款官方正版 IDE——Android Studio，功能强大堪比开发神器。Android Wear、Android TV、Android Auto 已经领先一步进入市场，越来越多的智能硬件都采用 Android 系统，希望借助 Android 生态环境来构建属于自己的市场。谷歌对于国内开发者也变得更加友好，全球最大的 Android 市场 Google Play 已经支持中国的开发者上传 App。本文作为 Android 这一年的总结，从系统本身、开发工具、硬件配置、国内外生态环境四方面介绍了 Android 这一年的发展与改变，并且结合当前市场大胆展望了 2015 年 Android 的发展方向。

系统

谷歌在 2014 年的 I/O 大会上发布了最新的操作系统 Android Lollipop，也就是 Android 5.0。Android Lollipop 是有史以来 Android 最大的一次改变。首先，在感官界面设计上，Android Lollipop 不仅使用了新的配色，同时使用了非常时尚的扁平设计，彻底迎来了 Android 系统的扁平化时代。此外，系统的多任务功能进行了一次基础性的重大调整。Android L 中用户将会拥有一个基于卡片的清单，其中呈现的并不是应用，而是任务。新的任务机制，能够节约大量的系统性能。另外全新的通知中心也不再乏味，当然还有大量的其它新特性，相信第三方系统插件的市场将会越来越小。

系统方面重重之重的改变应该是 Material Design。谷歌将 Material Design 定义为一种设计语言，其特点是能在将整个素材铺平的同时还遵循一定的物理材

质的需求。Material Design 的设计风格可以让应用感觉更活泼、具有更丰富的颜色，以及动画效果更真实等等。从技术角度分析，Material Design 完美解决了两个非常大的需求，其一是阴影，它所有的阴影都是默认系统实现的，开发者无需去自定义。另一个是动画，可以说 Android 5.0 将动画应用到了各个角落，动画效果甚至要超过 iOS，并且其效果不是简单的贴图，更像是真实的投影。谷歌自家的应用都已经使用了 Material Design，对于开发商来说，越早使用 Material Design，不仅有机会得到 Google Play 或国内市场设计推荐，更有可能提升 App 的留存率。

Android 5.0 对于硬件的兼容性比之前的版本有了重大提升，原生系统就支持多种设备，同时支持手机、Android Wear、Android TV、Android Auto，并且谷歌发布了这些设备的 SDK，Google Play 已经可以上传 Wear App 和 TV App。基于 Android 5.0 的整个生态链已经全部打通。

工具

一年前 Google 发布 Android Studio 测试版的时候，笔者天真的以为再有三个月就会出正式版，这一等就是一年多的时间。但这种等待是值得的，单从美观上评价，自带的 Darcula 主题炫酷黑界面实在是高大上，极客范，相比而言 Eclipse 的黑色主题太 low 了。Android Studio 亮相之初就支持 Gradle，Gradle 集合了 Ant 和 Maven 的优点，不管是配置、编译、打包都非常棒。Android Studio 的编辑器非常的智能，除了吸收 Eclipse+ADT 的优点之外，还自带了多设备的实时预览，这对 Android 开发者来说简直是神器。提示补全对于开发来说意义重大，Studio 则更加智能，智能保存，从此再也不用每次都 $Ctrl + S$ 了。熟悉 Studio 以后效率会大大提升。

Android 碎片化是让开发者非常痛苦的一件事情，一个 UI 需要去反复测试多个设备，Android Studio 解决了开发者的这个痛点，它支持多款设备的实时预览。Android Studio 还提供更多的特性比如内置终端、完善的插件机制，还可以安装 Markdown，你想要什么插件，直接搜索下载。Android Studio 自带版本控制系统如 GitHub、Git、SVN 等，可以直接 check out 你的项目。

硬件

在美国智能手机市场上，苹果可能是当之无愧的“王者”，但在全球范围内，三星的至尊地位则是无可置疑的。Android 设备包括高、中、低端产品，价格从几百元到五、六千元不等。从市场份额上看要高于 iPhone 系列。而国内市场用户最多的应该是小米，小米凭借 MIUI 成为了世界上最大的第三方 ROM 产商，迅速进入市值 100 亿美金公司队列。2014 年谷歌、三星、HTC、小米、魅族等等都发布了多款手机，这些手机的形态、尺寸、性能规格各异，大部分手机还无法升级到 Android 5.0 系统，Android 2.3 依然不死，Android 的

碎片化程度越来越高。对于开发者来说应用的开发更加困难了。不过我们不再有理由去抱怨，因为 iOS 的多屏幕适配繁琐性不低于 Android。

2014 年三月谷歌正式发布了针对智能手机和其它可穿戴设备的全新平台——Android Wear。Android Wear 除了最基本的查看手机通知消息，以及记录用户运动情况以外，还可以通过 Google Play 市场下载应用实现很多用户意想不到的功能。Android Wear 的发布催生了一大批 Android Wear 设备的诞生，年末苹果发布了自己的智能手表，微软同样推出了自己的智能手环。虽然 Android Wear 抢先一步进入市场，但却没有激起太大的浪花。与苹果发布的 Apple Watch 相比，Android Wear 无论从外观上还是从软件功能上都要逊色很多。笔者认为未来还会是苹果引领智能手表市场。

但 Android Wear 的表现并不影响 Android 进入客厅的步伐，乐视 TV、小米电视、天猫盒子已经展开了激烈的竞争，搭载 Android L 系统的 TV 将会是市场的主力军，传统电视厂商都已经开始研发自己的智能 TV，笔者想说的是 Apple TV 竞争非常激烈。

2014 年的硬件产品不全是美好，还充满了各种遗憾。小米没有推出更具吸引力的产品、魅族预售后 2 个多月都拿不到真机、智能汽车还不是真的智能...但对笔者来说，最大的遗憾应该是 Google Glass 消失了。2013 年 Google Glass 被称为是最伟大的发明与改变，那时我们都感觉世界如此之小，每一位极客都希望能体验带着 Google Glass 去环游世界。但 2014 年，Google Glass 没有新的进展，I/O 大会只字未提。有可能是因为体验的问题，有可能是因为材料问题，有可能是用户失去了耐心，Google Glass 发展令人担忧。

生态

谷歌在 I/O 大会推出了最新移动操作系统、设计语言，并正式启动 Android“连接世界”战略，将 Android 带入汽车、客厅、可穿戴设备、健康管理等更广阔的领域，谷歌利用自己的开放优势，借助三星、华为等第三方厂商正打算抢先在所有智能设备终端上布局。任何的设备，甚至灯泡都可以烧录 Android 系统，每个公司甚至个人都可以利用 Android 系统定制自己的智能硬件，然后利用已经成熟的应用生态圈去带动产品发展。

对于 Android 来讲，这样的好处是发展迅速，但不足是产品质量很难保证，碎片化严重。而苹果是软硬件追求极致的公司，无论是 iPhone、Mac 还是 2015 即将上市的 Apple Watch 都堪称极客产品。谷歌显然也意识到了这个问题，从去年谷歌的一系列举措，我们可以发现谷歌将加大对于 Android 的控制，软件层面将会出现更多的谷歌原生应用，而硬件方向，谷歌将会挑选一些大的厂商合作推出相关产品，比如有消息称谷歌将会与国内电视厂商合作在春节时推出 Android TV。

谷歌正在大步前进，连接整个世界，小米搭着 Android 的肩膀，已经成为了巨人，微信已经成为世界上最大的 IM，老罗也火了一把。前两年有人说，很少有 CP 从 Android 身上赚到钱，只愿意开发 iOS App。2014 Google I/O 大会上谷歌公布了一组数据，原话是这样的“In fact, since last year's I/O, we have paid out over \$5 billion to developers on top of Google Play”，比 2013 年的 20 亿美金翻了 2.5 倍，这还仅仅是 Google Play 一个市场，从而可见 Android 市场潜力有多大。国内有大量的公司依靠 Android 不仅赚到了钱，而且抢到了市场份额，比如像各种 App 分发市场。Google Play 已于 2014 年支持中国开发者，国内的开发者终于不用冒着被封号的风险去使用淘宝上购买来的 Google Play 账号。但反观国内市场，整个移动领域的盈利模式还令人担忧，很多开发商还停留在刷数据、冲 KPI 阶段，移动广告公司扣量严重、积分墙无法通过市场等等。如果无法赚到钱，投资人还会有多少耐心？

市场前景一片好，但小的团队、个人开发者的发展却变的更加的艰难。移动领域已经度过了发展最快的几年，市场需求已经基本饱和，越来越多的资源向大的 CP 靠拢。两年前一个技术、一个产品经理开发一款 App，迅速拥有 20 万用户的现象已经很难出现。很多渠道都需要 Money，App 市场也开始搞竞价排名，更是使新入市、低实力的开发者毫无竞争力。市场刷榜从 iOS 感染到了 Android，打包党变的更加猖狂，目前市场上有将近 70% 的 App 都被二次打包，笔者身边有一位朋友，辛辛苦苦开发几个月的 App，上架才一周就被打包，并且自己的 App 说是盗版被下架，而真正的盗版却成了正版。国家对 Android App、分发市场的监管力度目前还比较小，有好处但坏处也不小。国内外关于 Android 的技术环境还是相对薄弱，这与 Android 开发门槛低有很大的关系，国内的开发者社区活跃度比较低，优秀的技术分享更是非常稀缺。iOS 的技术氛围明显要强与 Android，iOS 有 objc.io 这样世界一流的开发社区，很多优秀的技术博客产出了大量的高质量文章。如果有能力，组织一个高、精、尖的社区应该有很大的发展潜力。

2015

由于 Android 天生安全性没有 iOS 高，随着手机变成银行卡、支付宝的钥匙，移动安全将更加重要。移动安全一方面是基于手机用户，另一方面是保证大量 CP 的正版利益。开发一款 App 的成本越来越低，国内早已经出现类似于一键建站的站长工具，所以对开发者的要求更加严格。目前入驻移动领域的厂商数不胜数，面向 CP、面向企业级的服务将会很受欢迎，2015 年一定会出现更多类似于 Umeng 分享、ShareSDK、可集成 IM 这种第三方服务产品，这个市场竞争还不太激烈，可做的模块很多。Android App 将会向模块集成方向发展。

苹果的 CloudKit 提供了完善且有弹性的后端解决方案，其目的是帮助开发者减轻编写服务器代码和维护服务器的需求，从而降低开发 iOS 应用的成本，有助于维护 iOS 生态圈的繁荣。谷歌 2014 年 10 份收购了 Firebase，Firebase

与 CloudKit 属于同一款产品——BaaS（后端即服务：Backend as a Service）。BaaS 因是移动互联网才诞生出来的工具。如今 O2O 发展的势头猛烈，很多传统行业本身的核心竞争力不在技术或者 app 层面，移动应用只是为了承载核心服务。所以能够以一种简单的方式搭建一个可用的 App、轻应用、HTML5 页面是最好的选择，就像我们不需要自己去搭建一个推送服务器，不需要自己去做数据统计、数据分析一样。2015 年，更多的 BaaS 产品将会出现，移动产品中的的数据存储、文件管理、消息推送等服务将会直接由 BaaS 产品提供，大量移动开发者不再需要去搭建服务器，开发成本将大大降低。

相信 APICloud、bmob 之类的平台会迅速成长起来，将提供更强大的功能。当然，BAT 也可能提供自己的 BaaS 产品，微信或许会为降低开发门槛提供类似 Parse 的服务，阿里借助 BaaS 产品给开发者提供新的变现方式也是很有可能的。

谷歌会将更多的资源开放给中国市场，并且会和国内的厂商合作，加强对 Android 的控制。无论是因为国内的市场透明度不好，还是因为国内市场已无发展潜力，都是时候开发国际版 App 了。猎豹就是典型的例子，凭借大量海外高质量用户成功上市。

基于 Android 生态的智能硬件、软硬结合产品越来越多，但暂时不会颠覆行业，进入普通消费者的生活还需要市场培养。



名额有限，报名从速

移动开发平台与工具应用实践

- ⌚ 时间：4月25日（周六）
- 📍 地点：北京国际会议中心



解读 2014 之安全篇：史诗级漏洞频发

作者 黄丹

编者按

2014 年，整个 IT 领域发生了许多深刻而又复杂的变化，InfoQ 策划了“[解读 2014](#)”年终技术盘点系列文章，希望能够给读者清晰地梳理出技术领域在这一年的发展变化，回顾过去，继续前行。本篇是解读系列的又一新作，互联网安全事件剖析，知道创宇技术副总裁余弦鼎力加盟话安全。

正文

初识余弦是在半年前，平时沟通算不上多，直到 [QCon 上海 2014 大会](#) 时，我们才渐渐熟识。他作为隐私安全专场的讲师，带来了主题为《程序员与黑客》的演讲，制造了一场惊心动魄的思维 PK，俘虏了众多“程黑”粉。这位看上去安静的美男子有着一颗并不安分的心，对这个世界充满好奇，最擅长的黑客攻防，在他的黑客生涯中发现过众多安全漏洞，实战经验极其丰富，经由他编写的《[Web 前端黑客技术揭秘](#)》一书也十分畅销。

2014 年是互联网严重漏洞集中爆发的一年，如 OpenSSL 的心脏出血（Heartbleed）漏洞、Bash 破壳（Shellshock）漏洞、Windows 的一些 0Day 漏洞等，受影响的企业、网站、硬件设备等范围很广。这一年对余弦来说也是不平凡的一年，他亲身参与了携程信用卡盗刷、心脏出血、破壳漏洞、12306 撞库等大事件的一线支撑。本文以时间为主轴，将这些事件串联起来，带大家回顾一下安全领域发生的几件大事儿。

携程信用卡泄露门

2014 年 3 月 22 日，中国在线旅游巨头携程公司被爆出“支付漏洞”，用户信用卡信息有可能被黑客读取，一时间引起成千上万用户的担心，纷纷打听是不是要更换信用卡。余弦认为这只是部分媒体对此事件的报道过分夸大，造成闹剧化趋势，事实上并没有那么可怕。关于漏洞产生的原因，余弦解释到：“按理说一个金融支付类的网站，应该遵守 PCI DSS（支付卡行业数据安全标准），这个标准体系里面有严格的规定，安全要怎么做？不能存，不能记录。如果记录了没有被曝光也许没事，但是被曝光出来了，又和人的财产相关，所以引起的动荡面还是很大的。当然，携程也迅速应急这件事，当一个企业成为舆论焦点的时候，携程做得比较好的是他承认了这件事，而且会改进这件事，并且去理赔，还是尽到了应尽的责任。”



OpenSSL 心脏出血

2014 年 4 月 7 日，OpenSSL 发布了安全公告，在 OpenSSL1.0.1 版本中存在严重漏洞（CVE-2014-0160）。OpenSSL Heartbleed 模块存在一个 BUG，问题存在于 `ssl/d1_both.c` 文件中的心跳部分，当攻击者构造一个特殊的数据包，满足用户心跳包中无法提供足够多的数据会导致 `memcpy` 函数把 SSLv3 记录之后的数据直接输出，该漏洞导致攻击者可以远程读取存在漏洞版本的 OpenSSL 服务器内存中多达 64K 的数据。每次可以读取 64K，循环去读，基本可以把这个机器内存里面保留的东西全部读出来。有很多用户名、密码，包括用户登陆的操作，收发邮件等私密明文信息。



余弦谈到 4 月 8 号开始，他们团队就开始应急，给国家包括很多媒体做了一线的支撑。大概 6 个小时后，了解到全球有 240 多万服务器受影响，覆盖面包括了使用 HTTPS 这样的安全网站，还有一些安全邮件的传输，还有 VPN 等。从全球来看，这个应急速度是非常快的，通过微博、微信等方式的传输，全球很多黑客都在行动，各种刷数据。总体来说，这次事件是整个中国安全应急的一次胜利。

Bash 破壳（Shellshock）漏洞



9月25号，继“心脏出血”漏洞之后，安全研究专家又发现了一个危险级别为“毁灭级”的漏洞——Shellshock 漏洞。这项漏洞的威胁程度堪比“心脏流血”，甚至影响更大。一方面是因为 Shellshock 所影响的 Bash 软件，同样被广泛应用于各类网络服务器以及其他电脑设备。另一方面，Shellshock 本身的破坏力却更大，因为黑客可以借此完全控制被感染的机器。

回忆起当时的情景，余弦不禁感叹道：“又是一个不眠夜！”。这种世界级的漏洞，很有可能半年之后又会发生。就好像渔夫懂得暴风雨要来临一样，黑客也有这样的预感，当时他就是这样的感觉。当然这种感觉并非没有道理，而是根据很多的线索感知出来的。破壳爆发的时候，国外的漏洞库 CVE 有一个比较专业的评价，他们有一个平衡的标准，心脏出血定义是 5 分，破壳漏洞是 10 分，顶级的。心脏出血是心脏出了血，破壳是“壳”破掉了，这都比较形象。因为它影响的是 Bash，翻成中文就是“壳”。

这几个漏洞都比较有代表性。这些漏洞的问题在于这个 Bash，它在 Linux 世界存在了 25 年之久，这个漏洞也存在了 25 年，是史诗级的。虽然不能说全

球所有的 Linux 都受影响，但基本上都受影响了。这时候就要去证明这个东西，像心脏出血，第一天看出 240 多万出问题，破壳一天有多少？通过研究发现探测的方式很复杂，不像心脏出血，针对它的服务器发几个包就可以知道情况。破壳不一样，在网络上，不同的设备、组件，所使用的 Bash 方式不太一样。破壳既可以远程也可以本地。远程方式比较复杂，需要 CGI 作为一个入口，并且这个 CGI 与 Bash 命令进行了交互，这时需要根据不同的设备，针对性地添加探测规则。但是一旦探测成功，就可以轻易入侵，在实测中发现几乎所有设备都用 root 权限启用这个 CGI，导致可以完全控制服务器权限。这些量级，多则几十万，少则几千。这个事情对整个业内的影响非常之大。

索尼影业被黑



11月27日，索尼公司被黑客攻击内网之后，宣布旗下 5 部电影遭到了泄密。这并不是一部影片遭遇泄密，而是大面积的泄密事件，足以引起绝对的重视。这次的事件全球的报道还是比较多的，各种花边的消息，各种说法都会有。据余弦透漏，索尼以前也出现过这样的问题，以前也得罪了黑客。现在又出现这样的问题，这个问题出现的还比较明显，有人告诉他是谁来黑他们了，他们这才发现，然后 FBI 进来调查，调查之后查到可能和朝鲜有关系。朝鲜前段时间因为这件事全国断网之类的，引起很多人的联想。当时八卦是说索尼那边有一个电影叫《刺杀金正恩》，惹怒了黑客，所以发生了这件事。而且这个可能还和内鬼有关，内外结合才会黑得这么透彻，还放出了很多索尼的内部资料，好几十 G，都可以下载。这就变得特别特别复杂、特别乱。背后牵涉到一些国与国之间政治上的东西，渐渐到经济上的一些影响、舆论上的东西。像索尼这么大的公司都会被黑客攻击，何况一些小公司呢？安全问题要尤为重视。

12306 遭撞库攻击

12月24日，漏洞报告平台乌云网出现了一则[关于中国铁路购票网站 12306 的漏洞报告](#)，危害等级显示为“高”，漏洞类型则是“用户资料大量泄漏”。据悉，此漏洞将有可能导致所有注册了12306用户的账号、明文密码、身份证件、邮箱等敏感信息泄露。对此，余弦认为：“12306，这本来就是一个很具有争议性的网站，尤其是在技术上。因为它和老百姓的生活息息相关，一旦出问题，当然是‘风口浪尖’，被各路黑客盯上也是很正常的事情。作为一个安全研究的团队，首先要想到一个好的思路，如何给出一个最佳、最严谨的真相？虽然很多人恨12306、骂12306，但是这个错误确实不是他们的。当12306的这种事情爆出来时，知道创宇安全团队很快开始做分析，经过很多的推敲，比如要随机调查、分析、审核，认为没有太多问题时，再对外发声，整个过程中团队的配合非常默契。”



对于这被泄露的131653条记录，当时有几种猜测，一种是撞库，还有是拖库，还有是用了第三方的插件（刷票）。从直觉来看，余弦认为是撞库出来的，然后去印证。如果是撞出来的，通过账号密码、身份证件能体现出来。当登陆12306网站测试时，发现身份证件等相关资料都是明文的，说明这个撞库的程序自动化是可以实现的。而且可以找到撞库的接口，通过这个接口，不需要验证码之类的，可以很方便的去刷。这些既然可以办到，那就说明撞库的可能性比较大。如果是撞库，那么哪些库可以作为依据来撞？于是将历史上泄露的一些库拿出来对比，发现相似度极高，大概有这样的一个依据。网上广为流传是由于第三方插件带来安全隐患，对于这一说法。余弦团队也进行了验证。他们从泄露的账号中找到几个和QQ号有关的，去添加测试。追踪是否使用第三方插件刷票的问题时，得到的答案都是否定的，那么可以排除通过第三方插件的可能性。对比之下得出了撞库这个结论。第二天12306的新

闻就出来了，此次攻击确实是因撞库引起的。年关将至，爆出这么一出，用户也真是醉了呀！

总结

以上只是 2014 年我们所认为影响比较大的网络攻击事件，实际上这还只是冰山一角，余弦认为，未来的网络空间将出现更多错综复杂网络袭击。比如说“工业控制”，尤其在国内，工业控制可以看作是民生的一个基础，水利、电力、石油、管道、燃气、交通、航天、化工、能源等全部都是工控。另外物联网、可穿戴设备、医疗设备等，这些都会在 2015 年爆发出来，安全伴随着不同的实体，这些实体在发展，安全同时也会发展起来。和人民的基础设备有关系的，一旦出问题了，带来的影响就不仅仅是数据丢失那么简单了，可能还会涉及到人身安全问题，迟早会有一篇新闻报道说，黑客控制什么什么，导致多少多少人死亡，这个也许会发生在 2015 年以后的某一年。

这么多安全事件引爆，都是黑客攻来攻去。所有这些事件的发生都和互联网的网络空间息息相关。这个网络空间如同人的大脑，网络空间里每一个终端的背后都是一个人，这个人给终端输入相关的指令，人的思维影响着终端的行为，终端的行为在整个网络空间传播，然后被存储下来。每一个终端都是一个节点，一个点一个点串起来，最终形成一个非常智慧的“大脑”，有存储、有记忆、有思维流通。它是群体构成的，不会依赖于个人的意志力去改变。

哪怕有再多可怕的事件发生，这都没关系。整个行业一直往前发展，人类的智慧力量无穷。一个黑客再邪恶、再凶狠，这个大脑都是可以治愈的。有受伤，治愈就 OK。未来还有很长的路要走，虽然压力无穷，但是正能量满满。

解读 2014 之前端篇：任性的颠覆式改变

作者 黄丹

编者按

2014 年，整个 IT 领域发生了许多深刻而又复杂的变化，InfoQ 策划了“[解读 2014](#)”年终技术盘点系列文章，希望能够给读者清晰地梳理出技术领域在这一年的发展变化，回顾过去，继续前行。

本篇是[解读系列](#)的前端篇，小编邀请到天猫前端团队的三七、铁军、不四、鬼道这四位专家来解读 2014 年前端领域最引人注目的几大热点。

HTML5 正式定稿，ECMAScript 6 规范从纸上走到现实

2014 年 10 月 28 日，W3C 宣布 HTML5 正式定稿为[标准](#)，这不仅仅标志着历经 8 年的标准纷争告一段落，也代表着 HTML5 这个名词会逐步洗去铅华，其技术真正融入到 Web 开发的每个角落，就像当年 Ajax 一样，当大家不再大张旗鼓吹概念和商业炒作时，正是其成熟时代到来了。HTML5 规范和以前最大区别是让 Web 最基础架构从 Web Page 升级到 Web Application，正符合主流互联网从桌面端迁移到移动端的趋势，是移动互联网终端碎片化的一剂良方。在移动智能终端性能和网络速度到达消费者需求时，将会是 Mobile Web 大展跨平台神威时，这从网络基础设施领先的韩国应用从纯 Native 到大量采用 Hybrid 形式就可以看出一些端倪。

HTML5 在尚未定稿前已有了大量实践，以至于其定稿之日也是成熟之时，同样下一代的 JavaScript 标准 ECMAScript 6 虽尚未定稿，但浏览器端 Chrome 和 Firefox 等新版本已实现 ES 6 的部分语法（Promise、Generators 等），同样在服务器端，Node v0.11 最新版本也已支持部分核心 ES 6 语法。这些 ES 6 新特性，大大地提升了开发者的效率。在 Node.js 中，已经有了非常优秀的基于 ES 6 特性的新框架，并已开始广泛地运用在生产环境中。

随着 Regenerator、6to5 等转换工具的出现，在前端使用 ES 6 的新特性也完全不是问题，这将大大提高整个 JavaScript 开发群体的效率，让大规模应用 Nodejs 的时代更快到来。

HTML 和 ECMAScript 是前端开发的基石，其快速升级和革新意味着这个领域面临的挑战和旺盛的生命力。

YUI 停止开发，小而美的 mv*库的流行

2014 年 8 月 29 日，Yahoo 宣布停止开发 YUI，如 Julien 在该博文所说，这几年前端行业越来越活跃，新技术和工具层出不穷，对于大而厚的基础库越来越不适应业务的发展需求。与此同时，单页应用技术符合 Web 应用化的趋势，一方面随着业务越来越复杂，前端 API 能力越来越强，数据和展现结合也越来越紧密，另一方面，Mobile 的发展对 Web 人机交互体验有更高要求，效果上要交互体验极致到 Native 的程度，性能上要前端库的高效且粒度及灵活性精细化，这也是类似 reactjs 等新型 mv*库开始流行的一个原因。

类似问题在 jQuery 身上也挑战很大，过去小而快的优点在移动时代已没有优势，需要面对移动端新的极致人机交互体验挑战。阿里开源框架 KISSY 正在使其核心模块粒度更加细小灵活，对低级浏览器的兼容拆分，在 Mobile 等高级浏览器下加载更少的代码，这是应对这一趋势所必须做的改变。前端框架和类库是为了提升前端开发的效率和品质，当人机交互环境发生重大变革时，这些基础设施都必须敢于大胆提早顺势而变，否则只能被淘汰。

基于 Web Components 的跨终端组件快速发展

2014 年通过指令化/声明式调用前端组件的形式发展迅速：比如 Angularjs、Reactjs 及新晋的 vuejs 等各种热议和实践，其中一个特点是 Directive 的引入。[Web Components 规范](#)将组件定义使用标准化，这种标准化正式跨平台跨终端业务急需的，为前端开发方案带来巨大机会。2014 年 Google IO 上《[Polymer and the Web Components revolution](#)》介绍，2014 年北京 QCON 豆瓣的《[DOMO UI](#)》、百度的《[跨终端组件实战](#)》，都是基于 Web Component 的实践落地（DOMO UI 类似 Web Component）。

究其背后原因：一方面前端开发越来越富交互化，组件共享复用也越来越频繁，如何高效一致地使用是每个组件库需要解决的问题，而 Web Components 的到来让我们看到了机会；另一方面，Mobile 的高速发展，让前端开发不仅仅只面对桌面一个终端，更要面对 Phone、Pad 乃至 TV 终端，Web 和不同的 Native 开始混用，如何让 Native 代码也能像 Web 组件一样方便调用，就需要引入类似 HTML 之类得声明描述组件，而 Web Components 正式符合这一特性的原生标准，为组件的跨终端带来无限想象。面对消费者终端的碎片化，Web Components 会成为跨端 UI 模块化协作的基础。

目前天猫正在构建跨终端高品质 UI 体系 MUI，从设计到客户端和 Web 前端一起打造一套 UI 设计规范和模块化组件体系覆盖所有端的天猫业务，从 iPhone 到 Android Phone，从 iPad 到 Android Pad，从 Mobile Web 到 Desktop Web，还有 TV 等，实现任何标准的 UI 设计都能够快速覆盖全站，其背后技术思想之一就是 Web Components。

基于 Node.js 的前后端分离方案流行，同时社区和企业边协作边竞争使 Node.js 在稳定服务和创新发展中平衡

1989 年 3 月 12 日，Tim Berners-Lee 创立了 WWW（Word Wide Web），Web 的迅猛发展成为 Internet 上最重要的内容承载方式，以至于很多人会认为 Web 就是 Internet。亿万互联网用户催生无数的 Web 开发者和巨无霸网站，Web 的规模化促使了前端的分工，于是 2001 年雅虎有了全世界第一个前端工程师职位，此时前端专注于 HTML、CSS 和 JavaScript，后端专注于业务和数据，而数据（Data）和展现（View）结合部分由于成本较低和技术难度不高而分工模糊，大部分情况下这部分工作依旧是后端工程师在负责。

2007 年 iPhone 诞生，互联网全面向移动快速进化，各种系统和硬件配置的 Phone 和 Pad 兴起使得用户访问互联网的终端碎片化，导致互联网产品都需要一套数据（Data）多个展现（View），所以 Data 和 View 结合的技术难度和成本剧增使得这部分工作必须从后端向前端转移，前端负责客户端和服务端所有的 View 及 View 相关的 Control，后端负责业务逻辑和数据并以 API 服务的方式向前输出，这样前后端彻底分离，对于产品开发而言前端只需要控制 View 和标准化的 Data 服务，不存在后端了。

前后端分离技术的难点是在服务端的前端，这个领域一直被后端开发语言和思想所统治，对于本来就很稀缺的前端工程师在技能和工作量上提出太高的要求，导致进展不顺利，直到 Node.js 横空出世。Node.js 出现，不仅让前端工程师终于有能力自己为自己打造提高工作效率的工具，让前端工程师发挥程序上的想象力，也让前后端分离有了更好的选择，所以整个业界非常多公司在这方面尝试，有些甚至尝试使用 Node.js 完全取代后端语言，比如 Java。目前还处于风起云涌的初期，所以即使在同一个公司如阿里巴巴内部都很多类似尝试，比如淘宝的 Midway、支付宝的 iChair 和天猫的 Wormhole 等，主因是难点并不在于 Node.js 技术本身，而在于和原有业务服务体系对接和运维能力上，所以切入点很多且难以标准化，先多点尝试相互竞争，后续在基于实际方案的基础上进行合并统一是我们目前的思路。

天猫的首页已经构建在 Node.js 上，不仅经受了 2014 双 11 的考验且性能表现优异，目前正在把这个方案应用到天猫所有活动和频道页面，到 2015 双 11 会有相当多的流量运行在 Node.js 上，那将是激动人心的时刻。

Node.js 开始大规模使用和其逐渐成熟完善且社区非常有活力关系密切，但从七月初开始，Node.js 核心开发者与 Node 社区核心参与者认为 Joyent 管理下的 Node 开发进展太慢，且对于社区的需求响应不及时，开始与 Joyent 公司进行谈判，希望将 Node 源码从 Joyent 公司拿出来，放到 Linux 基金会下基于社区来进行维护。最终事件以 Node.js 核心开发者 fork 了 Node 源码，重命名为 io.js 结束。2015 年 1 月 13 日，[io.js](#) 发布 1.0 版本，同时，node 也将发布 0.12 版本。这事件对 Node 社区影响非常之大。首先，造成 Node

源码的开发工作停滞了三个月左右，其次，io.js 开发活跃程度已经大于 Node.js，且 io.js 和 Node.js 的发展理念不同必然导致之后两者会渐行渐远，但从长远角度来看，竞争虽然带来阵痛但有利于更好的产品出现。

2014 年初，Node.js 当时项目掌门人从 Joyent 离职，基于 npm 创立了 [npm, Inc.](#)，开始致力于 Node.js 的包管理平台开发和维护。之前 npm 属于社区维护性质，服务不够稳定，随着 Node 社区的发展壮大，npm 服务的稳定性越来越重要，因此 npm, Inc 的成立保障了维系 Node 社区最重要的基础服务设施的稳定性。2014 年底，npm, Inc 发布新官网，同时重新定义 npm, Inc 为 JavaScript 的包管理工具和平台。

此时 npm 已经拥有了接近 12 万个模块，超越了 maven 成为了最大的包管理中心。随着模块数量的急剧增加，企业使用 npm 的需求也越来越高，npm, Inc 开始将目标瞄准了企业版 npm 市场，现在处于邀请公司试用期。而早在 2014 年中，阿里巴巴内部的私有 npm 服务已经非常完善，现在已经有每月超过 300 万次的下载，服务于全公司的 JavaScript 程序员。所以，社区驱动了创新和快速发展，企业会推动服务稳定和健壮，两者相互协作和竞争会让整个生态更有旺盛的生命力。

Native 定义为前端的一部分，并开始深入融合

2007 年 1 月 9 日，iPhone 诞生，带来了整个人机交互领域的颠覆式创新，对于前端技术也有了颠覆式改变，初期甚至到了讨论 Web is dead 的地步，加速 Web 世界的危机感和积极向移动端转型，同时随着 Phone 和 Pad 的严重碎片化和整个互联网都从桌面转向了移动，直接导致移动应用内容的规模化和多元化及连接和整合整个世界，越来越发现仅仅靠 Native 本身是不够的，需要 Web 和 Native 结合起来才能够满足极致人机交互和规模化联通世界的要求，比如微信其实就是这方面的表率。

前端的工作就是为人机交互的 UI 提供工程化方案，当整个互联网向移动转移时，原来的 Web 体系和工程方案已不适用了，这就是为什么 YUI 会倒下，而 HTML 和 JavaScript 要快速地推出革新版本，同样 Web Components 必须满足移动终端碎片化的模块化方案才能高速发展，而 Node.js 的流行恰好迎合上前端分离前端工程师需要掌控服务端前端的趋势。

这一些也仅仅是刚刚开始，TV 和 Watch 等越来越多碎片化的终端进入到日常生活，前端的挑战也刚刚开始且前所未有，这是最好的时代。Web 是桌面时代人机交互技术方案的王者，但在移动智能终端时代目前无法及时满足新兴的人机交互能力。这非常类似 Ajax 到来时，HTML + CSS 已经无法很好地满足人机交互 UI 开发的需要，前端要快速掌握 JavaScript 一样，移动互联网时代，Web 前端工程师需要快速掌握 Native 开发能力成为跨终端的前端开发工程师，这不是抛弃 Web 转向 Native，而是把 Web 和 Native 结合起来，就像

当年 HTML + CSS + JavaScript 结合起来产生巨大的威力一样。这不是 1+1=2，而是 1+1>2 的问题，不是简单的技术领域扩充，而是真正的人机交互技术深入探索，前端技术方案从来都不是由稳定的单一技术所能解决的。现在 Native 开发规模越来越大已开始在探索类似 Web 的 View 发布机制和模块化依赖关系管理等等，而 Web 也正在探索弱网络或不确定网络性能、内存管理及硬件调用的技术方案，两者结合（Hybrid）对于我们的大规模平台化业务来说是事半功倍的正确方向。

ArchSummit2014 深圳大会，手淘 Android 负责人无锋分享的手淘 android 架构《[手机淘宝的客户端架构探索之路](#)》中提到“像 Web 一样轻盈的 Native App！”可见从 Native 同学的角度已开始考虑向 Web 融合，而在天猫我们定义：前端 = Web + Native，目前天猫已有 10% 的 Web 前端同学拥有 Native 开发的能力，预计不到 2015 年中会扩大到 50%。虽然目前有大量的 Hybrid 应用和使用类 PhoneGAP 的混合开发，但 Native 和 Web 的深入融合远远不够，尤其是发布能力和大规模协作的能力上，以及对于组件、性能等多方面的相互协作。

之前一直讨论 Native 和 Web 孰优孰劣，谁取代谁，但经过 2014 相信更多人已意识到这是个伪命题，真正评判一项技术的价值是在业务场景中，选择合适且面向未来的技术最重要，需要思考如何用技术为用户和业务带来价值，天猫也正在前进的路上，随着越来越多人同时掌握 Web 和 Native，两者的协作会更加深入和相互发展，并作为整个前端的范畴带来更多的技术突破、效率提升和极致体验，而原来的 Web 前端工程师也会进行技能升级，勇敢地打破自我的壁垒拥抱移动端，尤其是 Native 技术，前端 = Web + Native。真正成为跨终端的前端工程师。

对于天猫前端而言，在新的一年里，Mobile 会变革为主场，主要有三个很明确的方向：跨终端组件、大规模 Node、Native 和 Web 融合。

跨终端组件 MUI： MUI 是天猫统一的跨终端 UI 组件库，这是设计师、Native 开发和 Web 前端一起协作的全站性质项目。之前已经历了两个版本完成了基础视觉规范和 JS 组件规范及管理升级机制，新一年的重要方向是：跨终端。目前正在 MUI3.0 进行的核心是天猫内部称作 FEModule 的组件体系，就是一个完整的组件规范（包括样式、脚本、模板和数据定义），实现前端、后端一致的组件体系，即一个模块完全独立，加上数据即可渲染，模块既可以前端渲染又可以后端渲染。MUI3.0 会基于 Web components 和 Native 组件融合规范，实现跨终端的组件体系。

大规模 Node： 首页在天猫双 11 中在稳定性及性能上已经被证实表现出色，同时对于前后端分离核心的数据 API 定义也有了系统化的规范和工具。新一代渲染引擎 Wormhole CDN 3.1 全网发布，支持 feLoader / feModule / 全局头尾，至此天猫应用、CDN、频道页环境的模版渲染环境基本都已经完善，Node.js 在天猫承担更多前端业务的时代已经到来。

Native 和 Web 融合：2014 年我们在技术和组织结构上做了很多突破，尤其是组织上把 Web 和 Native 前端调整为以业务维度的一个团队，前端 = Native + Web，持续推动团队转型深入到 Native。我们要把 Native 的高性能和系统能力同 Web 的发布能力和规模协作结合起来，这其中也有 Native 和 Web 互调的 Hybrid API，利用 Native 的缓存和系统能力把 Web 的基础打开速度做到 Native 一样的通用方案等等。

智能移动终端带来人机交互变革不仅仅导致了前端开发这个职位需要自我突破革新、重新审视和定义，更导致 UI 设计师的设计场景发生翻天覆地的变化，从单一的鼠标键盘大屏幕变成了多终端的触屏声音陀螺仪传感器等，设计需要更透析这些新的人机交互形式和技术才能够面向未来。新的一年里，三七将开始负责天猫的 UED 团队，把设计和技术结合起来，就像 [D2 前端技术论坛](#) 理念那样“好的设计驱动技术创新，好的技术给设计无限想象”，MUI3.0 就是设计、客户端开发和 Web 前端结合的产物，但这只是开始，三七如是说，未来还将继续颠覆、成长、蜕变。

2014 年 JavaScript 回顾

作者 李刚

2014 年对于 JavaScript 及其在 Web 上的地位来说是令人印象深刻的一年。我们已经在 [InfoQ](#) 上介绍了很多相关的文章，内容涵盖了各种 MVC 框架乃至专用的小型类库。

更多的开发者已经通过使用 [TypeScript](#) 或者 [Flow](#) 将类型系统加入到 JavaScript 语法中。而 Yahoo 在 YUI 问世 8 年之后，选择了[中止 YUI 这一项目](#)。新的构建系统 [Gulp 取代了 Grunt](#)。Node.js 在年初经历了[项目负责人变更](#)的巨变，并在年底产生了 [io.js](#) 这一分支项目。当 WebStorm 和 Visual Studio 以不可思议的速度发展的同时，开发者也有了新的工具能够选择，例如 GitHub 的 Atom 和 Adobe 的 Brackets。下面将介绍某些最流行的框架在过去 12 个月里如何发展的。

AngularJS 在 2014 年经历了最大的发展痛苦。今年开始时，最新的版本是 1.2.X。对这个版本最大的抱怨主要集中在文档的缺乏。2014 年 10 月，版本 1.3 发布了。这个版本增加了许多新功能，修正了一些缺陷并提升了整体性能。1 月份，在盐湖城举行了一个专门讨论这个框架的新型会议 [ng-conf](#)。10 月份时，这个会议转到欧洲举行。团队在会议上首次正式谈及了 [AngularJS 2.0](#)。对于这个新版本，社区里出现了非常激烈的争论，原由在于新版本的巨大变化有些令人措手不及，以及对新的语言（AtScript）的各种看法。

Ember 1.3 在年初发布了，在这个版本中加入了延迟路由特性。年底发布了[版本 1.9](#)，加入了对 Handlebars 2.0 的支持。2014 年 3 月底在俄勒冈州的波特兰第一次举行了专门讨论 Ember.js 的会议 [Emberconf](#)。

自从 2014 年 2 月 20 日 Backbone.js 的版本 1.1.2 发布后，就再没有任何官方消息了。虽然 GitHub 的代码库依然会持续收到提交请求，但是该框架的开发速率显得有些停滞不前。12 月中旬，第三次、同时也是最后一次 [backboneconf](#) 会议举行了。这次会议上，Backbone.js 的创建者 Jeremy Ashkenas 在主旨发言中讲到，Backbone 1.2.0 版本很快会发布。

2014 年是 ReactJS 获得巨大发展的一年。在 2 月 [版本 0.9](#) 发布了，目前版本是 [0.12.1](#)。在 2015 年 1 月，第一次 ReactJS 会议将在 Facebook 园区举行。自从 2013 年 5 月第一次公开发布后，开发者对它的兴趣在[持续增加](#)。

2015 年，JavaScript 上可能会出现更多混乱。支持 ECMAScript 6 的浏览器会增多，开发者将在最新功能的使用和浏览器兼容性之间的平衡上面临[艰难的抉择](#)。我们将会在 InfoQ 上进行追踪报道。

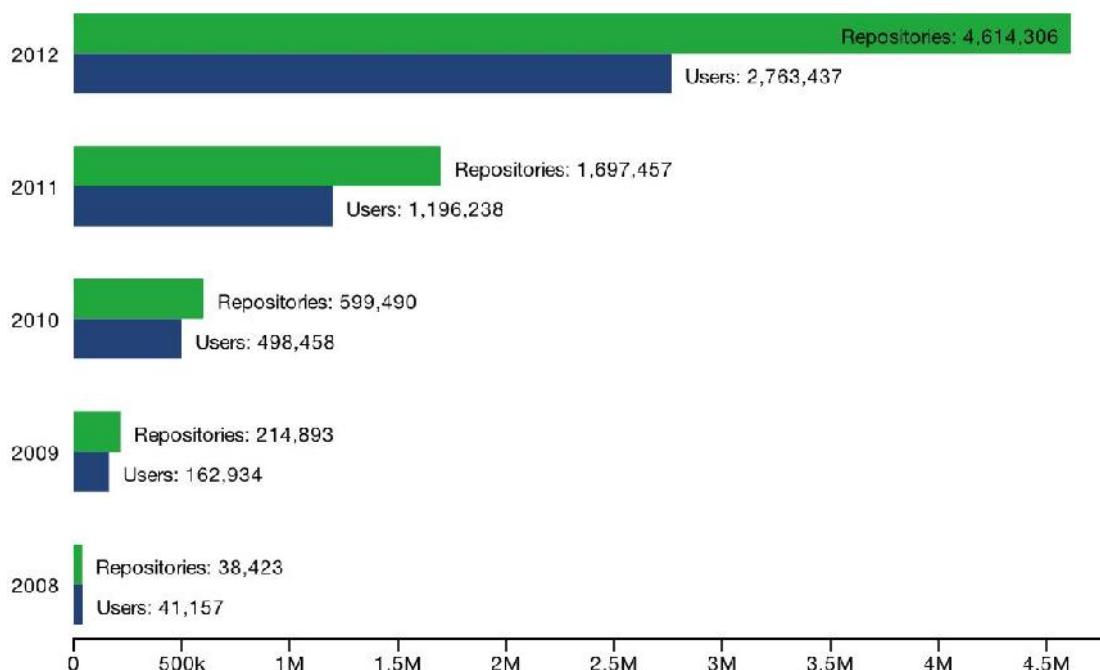
GitHub 中国开发者年度分析报告

作者 郭蕾

近日，GitHub.info 以 [GitHub API](#) 为基础，结合 GitHub 用户的地理位置信息，整理发布了《[2014 年 GitHub 中国开发者年度报告](#)》。报告中分析了 GitHub 中国用户的比例、活跃用户数、活跃时间段、仓库数量、PR 数量、项目 Star 数量、组织等信息，并就相关指标与美国开发者做了对比。同样，GitHub 官方也会发布[年度的用户报告](#)，InfoQ 编辑结合官方与社区发布的数据，对 GitHub 上的中国开发者情况做了一个简单的回望。

1. GitHub 用户数

GitHub 成立于 2008 年，是一个社区型的代码协作平台。根据 [2012 年的报告](#) 显示，GitHub 在 2008 年就有 4 万多的用户，在 2009 年迅速增长为 16 万，根据官方发布的增长数据可以推断出 2013 年已经有近 600 万（具体数字是 5843193）用户，新增用户有 300 万之多。GitHub.info 社区通过 API 计算得出，截止到 2015 年 1 月 20 日，GitHub 用户数已经超过千万（10475867）。如果这个数据准确，那 2014 年 GitHub 的用户增长就有 400 万。



从 2012 年的报告中得知，GitHub 中用户数最多（官方给出的数据是访问量最多，这里姑且认为访问量和用户数成正比）的五个国家分别是美国、德国、

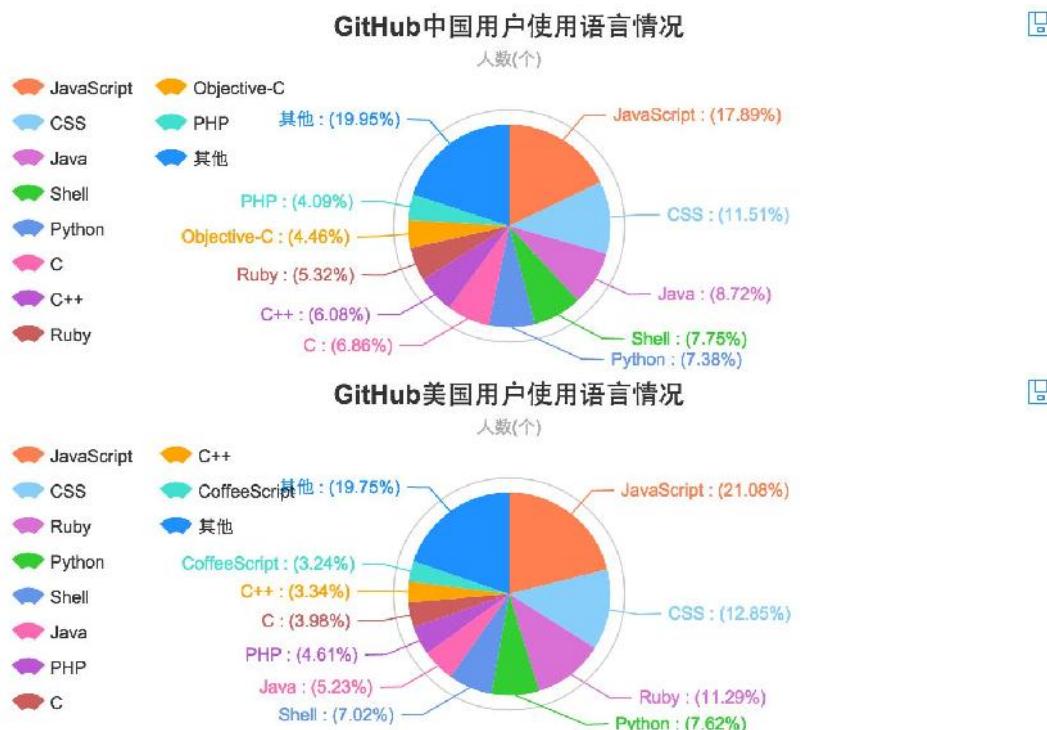
英国、中国和日本，2013 年是美国、德国、中国、英国、印度，而据[微博](#)上一位 GitHub 员工透漏，2014 年中国已经成为 GitHub 用户数最多的第二大国家。2013 年官方公布的数据显示，所有的注册用户中中国用户占 5.8%，如果以此比例来计算，那现在 GitHub 上的中国开发者大概有 60 万。

2. 开发语言

2014 年 5 月就曾有分析机构[利用 GitHub 来统计分析编程语言的发展趋势](#)，分析结果显示 GitHub 上主流的五种开发语言分别是 JavaScript、Ruby、Java、PHP 和 Python，其中 CSS 占的比例也比较大，C 和 C++ 处于中等水平，Go 之类的新型语言体量还是比较小。GitHuber.info 对中美开发者的语言情况做了对比，其中 JavaScript 和 CSS 占绝对优势，两者加起来接近 1/3，这也不难理解，JavaScript 和 CSS 是前端开发的必备语言，更何况还有 Node.js 之类的服务器端 JavaScript 语言。

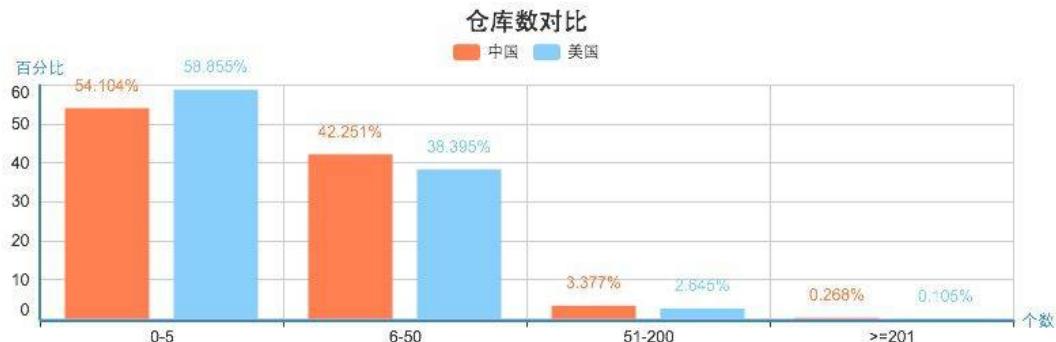
而排名第三的语言中美国和中国分别是 Ruby 和 Java，紧接着是 Shell 和 Python。值得注意的是，PHP 并没有进入前五，这也可能和该语言的使用场景有关（见过最多的 PHP 类开源项目就是各类 CMS）。两个国家的开发者中使用.NET 的用户都比较少，这和微软一直不鼓励开源的社区生态有关，随着新一年微软在开源方面的投入，相信.NET 相关的开源项目会逐渐增多。

需要注意的是，由于 GitHuber.info 统计国家时只能根据用户的 Location 信息识别，而很多用户都没有填写 Location 信息，所以文中涉及到 GitHub 方的数据时均指带有 Location 信息的用户。

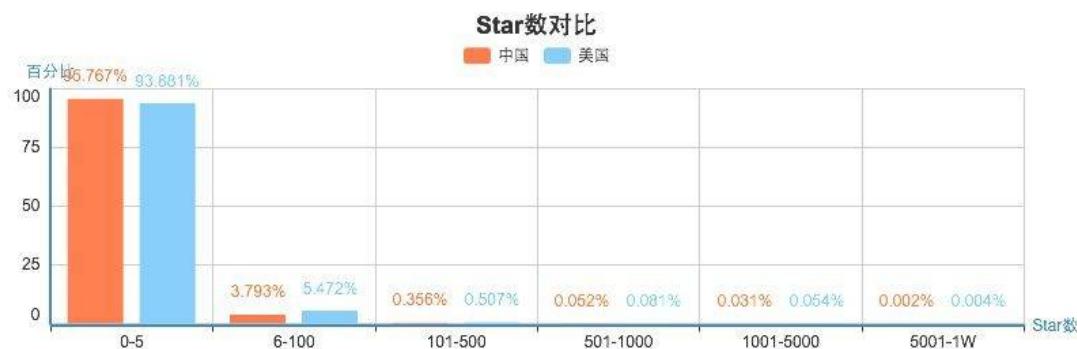


3. 项目情况

从仓库数来看，大多数开发者（75%）都拥有 0-5 个仓库，中美开发者差别不大。由于统计的是去除 Fork 的仓库，所以 0-5 个也属于正常情况，一个用户能玩转属于自己的几个开源项目，已经非常不错了。



而考量项目的质量可以从项目的 Star 数量来看，同样，绝大多数的项目都只有 0-5 个 Star，其中中国开发者的比例为 67%，美国开发者为 80%。如果以 Star 大于 1 万的标准来衡量优质项目，那中国没有，Star 数量最高的是 [awesome-python](#)，有 9393 个 Star。

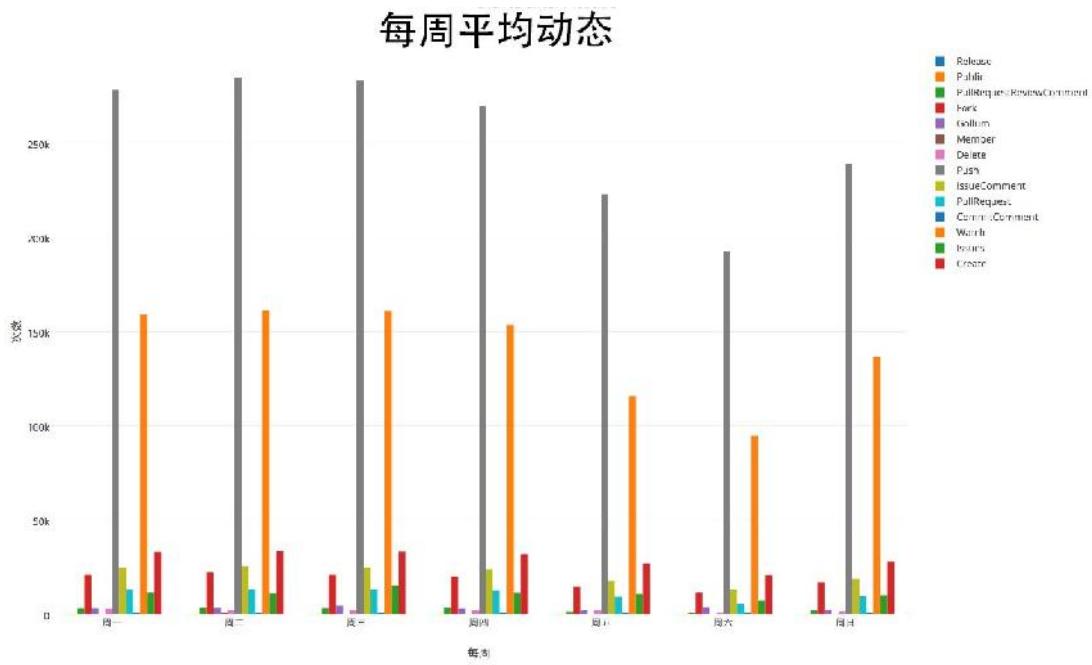


从项目的代码量来看，中美开发者悬殊比较大，GitHub.info 统计到的数据中美国用户的代码数量（字节）大多在 1 万到 10 万之间，而中国用户大多在 0 到 1000 之间（也就是基本为 0），美国开发者的代码量远大于中国。从分

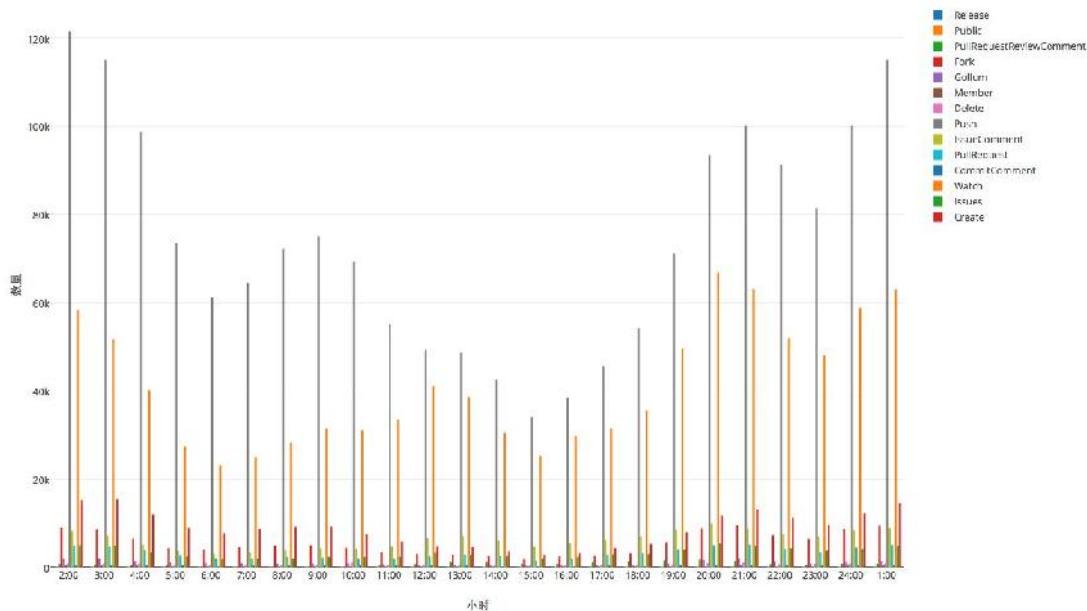
支数来看，美国和中国的对比相差不多，大多项目的分支数量都在3个以下，这也反映出部分项目还是纯个人业余开发，并没有达到需要规范分支的程度。但同样由于美国开发者基数比较大，所以总体来看，美国开发者好于中国开发者。另外，90%的项目的贡献者不超过5个人，90%的项目几乎没有PR和Issue，所以在人力投入比较少，且没有反馈的情况下，大多数的项目都没有长远的发展。

4. 活跃时间

GitHub.info 官方没有列出美国开发者的活跃时间，这里将官方 2013 年的数据作为对比，与美国开发者一样，一周中中国开发者最活跃的是周二和周三，最不活跃的是周六。比较有意思的是，周末两天美国开发者的活跃度都比较低，而中国开发者周日非常活跃，甚至比周五的还要高。细化到每天，更是伤人心。中国开发者每天最活跃的时间是凌晨 0 点到 2 点之间，而美国开发者是上午 9 点左右和下午 1 点左右。总体来看，代码推送的活跃度上午明显高于下午，美国开发者是白天明显高于晚上，而中国开发者是晚上明显高于白天。



日平均动态



5. 组织

组织是 GitHub 新的账号管理模式，旨在满足大型开发团队的需求。组织是非登录账号，需要以 GitHub 用户身份登录，然后再创建自己的组织。截止 2015 年 1 月 20 日，GitHub.info 根据 API 统计到一共有 2500 个中国区组织和约 25000 个美国区组织，GitHub 的用户数美国是中国的 4 倍，但是组织数美国却是中国的 10 倍，看来中国使用者并不喜欢团队协作。

组织部分总体的分布趋势和项目很类似，不过各项统计按比例来算要比项目信息稍差一些。按理说组织中的项目应该更容易吸引成员参与，从而在 PR、Issue 上有更好的表现。反过来说，目前中国的组织并没有充分发挥组织的作用，对开源项目的发展推动不大。目前国内的优秀开源项目主要还是依靠线下团队的合作开发，距离美国开源项目的众包模式还有些差距。

刚刚从 GitHub 官方邮件中得知，GitHub 今年不会再发布 [Octoverse 报告](#)，不过感兴趣的读者可以邮件 press@github.com 获取一些官方可以公布的数据，InfoQ 正在与官方联系，希望获取一些中国开发者的详细报告，我们也会及时公开各项数据。从 GitHub 用户增长数量也可以看到开源的趋势，随着国内开源环境的成熟，优质的国产开源项目也越来越多，InfoQ 也会全程跟踪报道国内的开源状态，敬请关注。

“技术帝”盘点之女神篇

全球软件开发大会

2015.4.23-25 北京国际会议中心



罗韩梅

腾讯数据平台部高级工程师

Gaia——万台规模的
Docker应用实战



吕婷

青云前端工程师

与产品紧密
前端工程师的成与责



Barbara Horne

The Business Channels咨询顾问

让商务智能成为主导



黄小微

蚂蚁金服高级技术专家

面向服务的质量保障



郭昱桢

蚂蚁金服前端工程师

全栈时代下的前端工程体系



刘艳凯

惠普首席云计算技术专家

明智之选，轻松云之路



房秀丽

搜狗高级经理

从“救火”走向“防火”



刘晓莹

蚂蚁金服应用架构师

互联网金融系统如何去O
——蚂蚁金服金融
核心平台经验实践



QCon

抢票热线: 010-64738142

会务咨询: qcon@cn.infoq.com

更多精彩内容 敬请登录: www.qconbeijing.com

促进软件开发领域知识与创新的传播

架构师 特刊

架构师 ARCHITECT

跨越2014，展望2015



解读 | Review

安全 云计算 前端
Docker Android iOS

分析 | Analyse

2014年JavaScript回顾
GitHub中国开发者年度分析报告

InfoQ 特刊



本期主编：崔康

流程编辑：丁晓昀

发行人：霍泰稳

读者反馈/投稿：editors@cn.infoq.com

微博、微信： @infoqchina

商务合作：sales@cn.infoq.com

本期主编



崔康 InfoQ 中国总编辑，致力于中国 IT 领域知识与创新的传播，目前负责 InfoQ 整体内容的品牌和质量，同时担任 QCon、ArchSummit 大会的总策划。技术人出身，毕业于天津大学计算机学院，在加入 InfoQ 之前，在某大型外企长期担任协作软件平台的技术负责人和架构师，在相关技术领域积累了一定的经验。从 2008 年开始参与国内社区的技术传播，在多家科技媒体先后发表过数十篇文章，并出版多本译著，总产量超过 50 万字。可以通过 tyler.cui@infoq.com 或者微博（“崔康 Tyler”）与他联系。