时刻关注企业软件开发领域的变化与创新

架构师

12月 ARCHITECT

特别专题

HTML5时代开端

当HTML5来敲门

Opera技术布道专家谢子斌谈HTML5

HTML5在豆瓣中的应用

使用HTML5构建下一代的Web

加速你的Hibernate引擎（下）

Tellurium自动化测试框架介绍

Visual Studio 2010扩展管理器

淘宝UED技术负责人小马畅谈Web前端开发

InfoQueue
Enterprise Software Development Community

每月8号出版

篇首语 | **Editor's letter**

这个冬天不太冷

每当进入 12 月份，空气中总是洋溢着节日的快乐气氛，圣诞、元旦，还有不久之后的春节让大家憧憬不已。在这样一个美妙的时刻，InfoQ 中文站出版的《架构师》电子杂志又如约与读者朋友们见面了。

在本期《架构师》中，除了过去一个月的文章和新闻精选，我们策划了 HTML 5 专题讨论。众所周知，HTML 5 不算是个新话题了，但是围绕着 HTML 5 的技术优缺点、与 Silverlight 和 Flash 之间的关系等等问题，最近在社区中掀起了一翻热浪。为了让读者更好的了解 HTML 5 技术，特别是现阶段如何处理 HTML 5 在应用中的实践等核心问题，InfoQ 中文站邀请了国内外 Web 前端技术领先的企业和专家就 HTML 5 相关的各种话题做了讨论，有人物专访、有深度分析。通过本期《架构师》，读者会看到 HTML 5 在豆瓣网上的先锋实践，还有支付宝专家对 HTML 5 应用的衷心建议，以及 Opera 布道师对 HTML 5 前世今生的娓娓道来。当然，这不是全部，InfoQ 中文站会陆续发布更多有关 HTML 5 专题讨论的文章，请感兴趣的读者朋友继续关注。

提到 InfoQ 中文站，我不禁想多说几句。在这个冰凉刺骨的冬天，InfoQ 中文团队内部却有一番热火朝天的景象。大家都在积极的想办法让 InfoQ 更好的为国内技术社区服务。多原创！读者朋友可能注意到了，最近 InfoQ 推出了技术专栏，请业界的翘楚编写某专题的技术文章，得到了读者的热烈响应。InfoQ 随后还会陆续推出其他原创专栏，遍布各个技术领域，总有一款适合你。还有即将到来的 QCon 2011 北京大会，目前还在积极的筹备当中，专家名单依然熠熠生辉，去年的盛况我还历历在目，希望到时候和大家继续相会在会场上，不见不散！

本期主编： 崔康

目　　录

[篇首语]

这个冬天不太冷 1

[人物专访]

WPS 技术总监杨钢谈 WPS 和技术团队 4

[热点新闻]

天涯新款 KEY-LIST 类型内存数据引擎--MEMLINK 10

APACHE 怒了，威胁说要离开 JCP 13

讨论：研发团队中最合适的开发测试比是多少？ 15

IPHONE 的开发环境优于 ANDROID 吗？ 18

HTML5 VS. FLASH：ADOBE 面向何方？ 20

[特别专题]

当 HTML5 来敲门 23

OPERA 技术布道专家谢子斌谈 HTML5 28

HTML5 在豆瓣中的应用 31

使用 HTML5 构建下一代的 WEB FORM 37

[推荐文章]

加速你的 HIBERNATE 引擎（下） 42

TELLURIUM 自动化测试框架介绍 57

VISUAL STUDIO 2010 中的扩展管理器 64

淘宝 UED 技术负责人小马畅谈 WEB 前端开发 79

[新品推荐]

IBM 发布 WEBSPHERE 应用服务器 V8.0 测试版 **82**

WSO2 发布 CARBON STUDIO **82**

ECLIPSE VIRGO 发布了 **82**

GOOGLE 发布了 MOD_PAGESPEED **83**

SPRING PYTHON 1.2.0 RC1 发布了 **83**

HIVE 0.6.0 发布了 **83**

[推荐编辑]

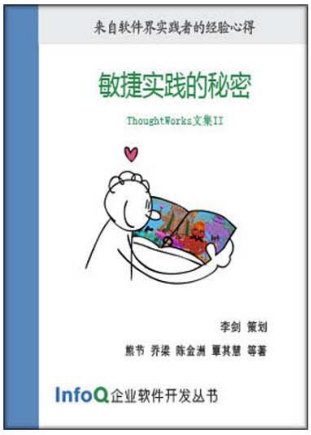
翻译团队主编 张龙 **84**

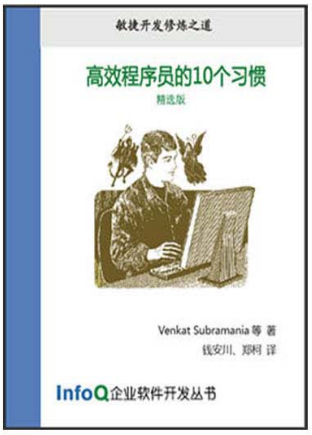
[封面植物]

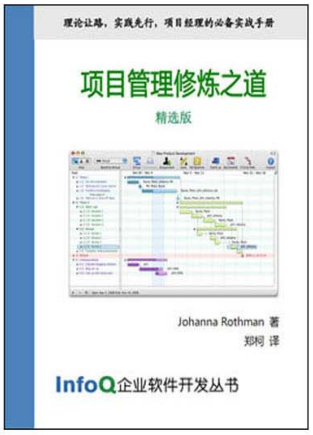
金花茶 **85**

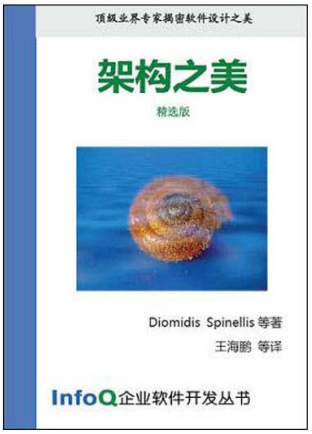
InfoQ企业软件开发丛书

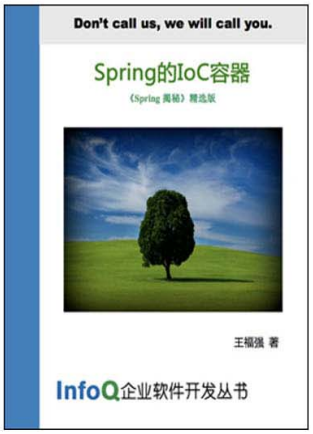
欢迎免费下载











商务合作：sales@cn.infoq.com　　读者反馈/内容提供：editors@cn.infoq.com

人物专访 | Interview

WPS 技术总监杨钢谈 WPS 和技术团队

杨钢，金山 WPS 的技术总监，一直带领技术团队从事大型软件的架构设计。我们本期《架构师》月刊就大型复杂软件的架构开发以及大型技术团队搭建和管理以及 WPS 的发展等方面刊出了一篇关于他的任务专访。

杨钢，金山办公软件子公司技术总监。2001 年加入金山软件至今，参加了自 WPS 2002 以来历届版本的开发。研究方向为大型应用软件的设计与实现，致力于将架构设计变成一个有理论基础，可复制的过程。

InfoQ：大家好，我是 **InfoQ** 中文站的编辑，今天来到我们采访室的是金山 **WPS** 的技术总监杨钢先生，杨钢先生刚才跟大家分享了非常精彩的演讲，现在首先让他做一个自我介绍吧。

杨钢：我是 WPS 的技术总监杨钢，我 2001 年加入 WPS，一直在 WPS 从事技术框架方面工作，我的主要工作方向是大型软件的架构设计相关的问题和大型软件的复杂度问题，我希望把大型软件开发的过程变成一个有理论支撑的可以重复的一个实践活动。

InfoQ： 如您所说 **WPS** 是一个非常复杂的大型软件，同时它用到了很多种语言，可不可以跟我们分享一下 **WPS** 在这种混合架构中遇到过的一些典型的问题，以及咱们是怎么解决的？

杨钢：这一点的话要分两部分来谈，首先是客户端的 WPS OFFICE，我们当时为了加快界面的开发速度，除了底层是全部 C++，因为可以获得更大的控制能力和更大的优化的可能性，在界面我们使用了 Delphi，因为 Delphi 是当时比较流行的构建界面的 IDE。在前期，我们很快就构建出来一个 OFFICE 的界面框架，但是随着这么多年 WPS 的不断开发和维护，我们也开始对当初的这个决定进行一些反思，因为在这种紧耦合项目中间，界面和内核之间的耦合度非常高，交互很频繁，数据结构也非常的复杂，在这种紧耦合的交互过程中，使用两种语言，遇到了很多的问题。首先是错误调试的问题，因为很多关于界面的地方，都需要在 C++ 和 Delphi 两边来回调试，目前没有一个很好的调试工具来做这种混合调试，我们也做过一

些辅助开发工具，但是毕竟我们不是做开发工具的厂商，在这方面事实上能做的比较有限；第二个关于程序员能力的培养，因为 C++是一个非常复杂的语言，要熟练、精通 C++的话，可能需要很多年的时间不断的积累。Delphi 它也是不吃素的一门语言，这两个合起来，无疑给我们整个开发团队的学习负担增加了不少，这是对于客户端的 WPS 来说。对于 WPS on WEB，这个 WEB OFFICE 项目，情况又发生了变化，我们在这个阶段把整个体系，铁板一块的客户端进行了重新的分解，我们从里面找出来紧耦合和松耦合的边界，对于紧耦合的边界，我们仍然还是比较主张使用一门语言。但是对于松耦合的边界，比如说 WPS 的内核模块和 WPS 的外围模块，甚至包括 WPS 的界面模块和 WPS 的内核模块之间，因为要做服务商和客户端这种分解之后，它不得不变成松耦合。那么这种情况下用混合语言，其实也是可行的，而且对于松耦合的边界来说，对于不同的部件使用最适合的语言是一个更好的方案。

InfoQ：您刚才提到了程序员能力培养和开发的要求，请问 **WPS** 团队使用的开发方式是敏捷还是传统的？

杨钢：这个东西可能不太适合一概而论，说实话我们从开发效率上来说的话，应该是属于传统中的敏捷，首先我们会有一个设计过程，我们有架构师，会把软件的细节定义的相对来说较为清楚之后，就进行软件的开发，这种相对于敏捷来说，它是偏向于传统的。但是对于传统的开发模式，特别是像日本软件企业，每一个函数的输入、输出都定义的，文档化很好的这种开发模式来说，WPS 又是一个相对非常敏捷的一个项目。在我这边管理的几个项目里面，我对他们的要求是只要有类图基本上就可以开始开发，那么对于我和一些项目的架构师来说，只要给我一张类图，基本上你的思路我都已经清楚了，除非你设计的是一团糨糊。

InfoQ： 您刚才说道 **WPS** 团队的人数目前已经有 **200** 多人，那 **WPS** 团队怎么样逐步成长到两百多人，在这之间有没有遇到过团队成长性的一些问题？

杨钢：其实在这一点上，我们做的有一点迫不得已，因为 WPS 经过了几起几落，落的时候人就很少，起的时候人很多。在起的阶段，会遇到一个很大的问题，就是大量新人的涌入。为了处理这个问题，我们采取了各种方式，首先是在金山集团下面会有一个集团训练营，这个训练营是独立于 WPS 之外的，它是为全公司服务的。他们会对新员工做最基本的培训，这个训练营的两位掌门人，事实上都是以前 WPS 的元老级程序员，董博、沈家政，都是第二代 WPS 程序员，所以他们为 WPS 培养人的时候更得心应手，他们在这个地方起到了非常大的功劳。然后 WPS 团队里，主要就是采取这种师傅带徒弟的方式，新人在经过了训练营之后，可能就直接参与项目，在项目中演练。我们其实给新人很多犯错的机会，我们在更大的尺度上去控制他就是不会造成乱子，但是在更细的尺度上，他可以不断的去试错。在这种可以让他充分去主动体会的基础上，新人的成长一般都比较快，实际上这也是很迫不得已的，

我们挺希望人员很稳定的。

InfoQ：想问一下 **WPS** 除了技术团队，还有其他的比如需求、产品设计的，或者是测试等相关的团队，咱们的技术团队跟这些其他团队，是怎么样一种合作模式呢？

杨钢：其实刚才给你提到的 200 多人是指的整个金山 WPS 研发部的人，包括了开发、测试和产品，目前 WPS 还是开发的人最多，但是测试和产品也是一个不断壮大的群体。因为在 WPS 里面，大家是为着一个大项目在活动，彼此间的界限很模糊。这种交互都比较顺畅，没有太多的流程，基本上都是在项目组里面互相讨论。

InfoQ：咱们推出 **WPS on Web** 是基于什么样的考虑呢？

杨钢：首先 WPS on Web 还在开发中，因为我们的目标是除了会提供文档的那种存储、管理协作这样的功能之外，希望我们的 WEB OFFICE 能够给用户一个和客户端 OFFICE 一样的体验，可能在大家的心目中觉得 WEB OFFICE 就是一个补充品，有时候临时性的用 WEB OFFICE 应应急，但是我们觉得其实那主要是因为 WEB OFFICE 还不够成熟，但是随着 WEB OFFICE 不断的成熟，它实质上有可能会变成另外一个选择，而不是说只是一个躲在后面的一个阴影。

那么 WPS 做 WEB OFFICE，说到坦诚一点，大趋势，别人都做，咱不能落后。但因为随着互联网技术的不断发展，特别是 RIA 技术这段时间的不断成熟，使的互联网 WEB 应用技术，已经足以支撑像 OFFICE 这样复杂的应用，在这段时间有一个很大的趋势，就是各种各样的应用都逐渐的从客户端向 WEB 形态开始迁移，也包括企业里面使用的一些系统，像财务，像 ERP 之类的，现在也越来越多的向 Web 方向迁移。Web 应用有一些比较好的特点，第一个是它的易传播性；第二个是对于企业来说，那个零部署成本。这些优点使的 Web 应该会成为下一代应用程序承载的主流平台。

InfoQ：这个是不是跟咱们的 **WPS** 客户端形成一定的竞争呢？

杨钢：我们内部也有很多这样的争论，我们对用户提供的价值，不仅仅是在用户文档的创作和编辑这个过程，当然这个是以前的 WPS 带来的。但是随着我们将更多的互联网服务元素加入整个 WPS 大家庭里面去，我们可能会涉及到文档创作之外的关于文档其他的流程，比如现在已经做到了的文档的存储、备份、管理、归档、分享。在整个活动中，编辑虽然是其中不可缺少的一环，但是它只是整个活动的一个部分，其实 WPS、WEB OFFICE 都是整个 WPS 办公服务的一个部分，是参与到整个办公服务中，都是不可缺少一环，并不是说谁要去替代谁。

InfoQ：开发 **WEB** 程序都会考虑到浏览器兼容性的问题，现在 **WPS** 的 **WEB OFFICE** 对浏览器的兼性支持的如何？

杨钢：其实浏览器兼容对于 WEB OFFICE 开发是一个必须要考虑的因素，这个也直接会影响到我们对 RIA 技术的选型，在刚才讲的时候，其实就有朋友关注，为什么 WPS 不用 Ajax，而是用 FLASH，其中我提到了两个原因，第一个原因是因为 Ajax 线上开发工具还不太成熟，对于一个超大型项目来说，可能是支撑不住，比如像 WPS WEB，我们目前粗劣统计大概有八十万行代码。然后我们根据我们的生产力估计到年底可能会有两三百万行代码出现，那么这么大的代码量，目前 Ajax 的应用是支撑不住的；还有另外一个原因就是跨浏览器的一个问题，根据我们的调查验证，目前 FLASH 在跨浏览器上面的问题要比 Ajax 要少一些。然后我们也和 ADOBE 在不断沟通这方面的问题，ADOBE 也在解决它。所以从目前来看，FLASH 跨浏览器目前不是一个很大的障碍，相反的 Ajax 多浏览器调试的成本会更高。

InfoQ：现在 **WPS** 同时有客户端版本和 **Web** 版本，在代码重用上有没有什么可以跟大家分享的？

杨钢：首先比如说像我们在线的代码采取的是一个客户端服务器的模式，就是我们会有很多的计算是在服务端进行，这些在服务端进行的代码，基本上都是在原有的客户端代码分离出来，然后进行适当的改造，这是很大的一部分代码重用。然后除此之外，我们会有一些计算是由客户端做，甚至包括 UI 部分、显示部分，必须是客户端做得。这一块对我们来说，大型软件最关键的是在于它的架构是否清晰，各个元素的分解和配合是否合理，耦合度是否低，这种架构设计方面是否可以重用是跟语言是无关的，在这个方面实际上是重用了 WPS 很多这种架构资产。

InfoQ： 对于 **WPS** 这样的大型软件，他对架构的要求是非常高的，大家对 **WPS** 一直都非常关注，如果想成为 **WPS** 架构成员之一的话，有什么要求？

杨钢：这个问题到是比较难，实际上，我们就以 WPS 典型的一个架构师的成长过程为例，首先就是 WPS 在招人的时候，我们完全不管学历或者是其他的认证，我们的招人标准之一就是要非常喜欢程序，然后非常喜欢去琢磨技术问题，包括我自己虽然现在担任 WPS 的总架构师，但我平时的程序代码工作量比一些项目经理还要多。我觉得，成为架构师的一个充分条件，就是他一定要对程序有大量的程序做接触，对程序有很强的这种感觉。然后在这个基础之上，在 WPS 里面，我们会进行适当的培养，让他开始从小模块的设计开始做起，在成为一个 WPS 所需要的架构师的这个过程中，他需要的进行一些培训，比如说怎么样用架构语言来表达一个设计，然后包括一些就是业界上很多这种，已经广泛成熟的一些技术的掌握，比如像类图的绘制，比如像设计模式的理解和熟练应用。那么其实和程序员的成长过程一样，我们的架构师也是在实践中慢慢的锻炼出来，会有一批人成长为一个系统架构师，但是有没有一个办法能够让你一定成为一个系统架构师，我觉得这点，我可能目前还经验不足，

还没有办法给出一个答案。

InfoQ：未来咱们 **WPS** 有什么长期的发展计划呢？

杨钢：其实 WPS 之前给别人的印象就是 OFFICE 的克隆品，反正这个我们自己也承认，你不克隆就没人用，以前的 WPS2000 很有个性的，很多老用户很喜欢，觉得我们丧失了自己的风格，但确实太小众了。我们的总经理张泉原来说过一句话，先继承后超越，我们觉得经过了前面几年的发展，我们已经过了继承这个阶段。从去年开始实际上 WPS 已经酝酿、布局、超越这么一个过程，首先我们对项目组进行了调整，把整个 WPS 原来一个客户端项目，分解成了包括客户端、办公服务、WEB OFFICE 等好几个项目群。然后我们在客户端、在网络服务上面就探索了很多跟文档相关的服务，除了刚才提到的文档的存储、分享之外，我们还有一些跟文档创作相关的，举个例子，在 WPS，打开的时候有一个首页，一个模板展示区，这个模板展示区里面有很多精美的模板，这些模板其实不是我们做的，是很多专业的设计公司，他们做了之后免费提供给 WPS 用户，我们走一个相当于是流量渠道这么一个模式，跟他们交换资源。

然后我们在 WPS 里面，提供了一种叫做扩展插件平台的机制，这样第三方的开发者可以给 WPS 开发大量的扩充功能的扩展组件，现在的组件不是太多，在 WPS 官网上应该也有，这个数目我确实不记得，但是有大量的开发者已经在做这个事情了，这些组件给 WPS 带来了很多微软 OFFICE 没有的功能， WPS 在办公创作这一块，也开始寻求在已有的成熟经验上，能够更好的让用户以更高的效率创作文档，比如表格是中国人用的很多的一种文档元素，但是微软 WORD 里面它的表格操作其实是很复杂，特别是加上了合并单元格以后， WPS 在这方面一直探索一些新的操作模式，去简化这些表格操作，有一些这种操作方式，后来微软也给学进去了，这是关于客户端。另外一条产品线是关于办公服务的，我们觉得在整个文档的过程中，文档的编辑只是其中的一小部分，因为从我们的统计来看，虽然我们 WPS 是有免费的个人版，有大量的个人用户，但是从用户的反馈，或者是一些渠道我们了解到，其实很少有用户用 WPS 去做像日记一样的纯个人需求，他们做的都还是和公司相关的事情，那么他这个文档做出来肯定是要给别人看的，肯定是有一些几个人合作写一个出来，有一些可能要有审批之类的过程，这些过程以前只有很大的公司花非常高的费用去实施一个 OA，然后才能将这些流程给电子化。

我们觉得，WPS 本身可以帮助企业在文档编辑的程度上，在这个层面上实现电子化，在文档除了编辑之外，其他的流转、共享过程中，我们也能够做让特别是没有这种很大的预算去实施 OA 的公司，也能够很方便的享受到这个服务，这个是第二个产品线，第三个产品线就是 WEB OFFICE，WEB OFFICE 的长远发展现在并不是很好精确的描述，但是我们都相信未来

它是一个大趋势。

InfoQ：非常感谢杨钢先生的精彩分享。

原文链接：http://www.infoq.com/cn/articles/yg-wps

相关内容：

- WPS：在网页中使用PostScript绘图
- 系统级复用的成功要素
- 我眼中的Visual Studio 2010 架构工具
- 一个技术观察者的年度展望
- 2010 年大规模技术架构的思路

InfoQ中文站

www.infoq.com/cn

我们的**使命**：成为关注软件开发领域变化和创新的专业网站

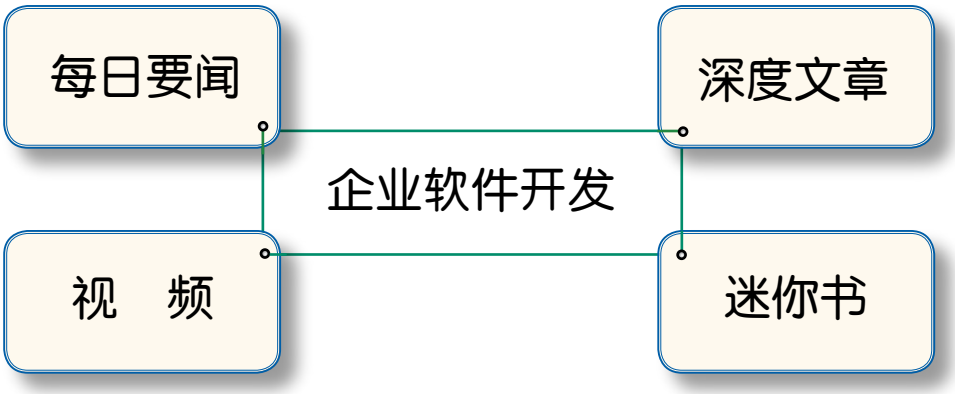
我们的**定位**：关注高级决策人员和大中型企业

我们的**社区**：Java、.NET、Ruby、SOA、Agile、Architecture

我们的**特质**：个性化RSS定制、国际化内容同步更新

我们的**团队**：超过30位领域专家担当社区编辑

……



讨 论 组：groups.google.com/group/infoqchina
编辑电邮：editors@cn.infoq.com
广告合作：sales@cn.infoq.com

热点新闻 | News

天涯新款 key-list 类型内存数据引擎--Memlink

作者 张凯峰

天涯社区最近开发了一款数据引擎——Memlink，并将其开源。对于为什么会出现这样一款开源项目、它的能力和市面上的其他款同类型项目相比有怎样的优势，InfoQ 中文站特地采访了天涯社区在北京研发中心的技术负责人冯勇先生。

InfoQ：您好，能请您先自我介绍一下吗？您最近在做哪些有趣的事情呢？

冯勇：大家好！我是天涯技术中心系统平台部负责人冯勇，系统平台部是今年刚组建的部门，旨在优化天涯线上产品的系统架构。天涯是一个有十二年历史的网站，对于一个累积了十二年补丁的系统进行重构、优化，本身就是一件很有趣、很有挑战的事情。

InfoQ：是出于什么初衷，天涯会开发出这样一款数据引擎出来呢？并且最后要开源出来。

冯勇：近些年，Nosql 系统非常流行，也确实对 sql 系统进行了合理补充，为 Web 应用提供多种数据解决方案。但是在开源 Nosql 系统中，key- value 系统可选择较多，而 key-list/queue 系统可选择较少，因此我们开发了 memlink 来满足我们自己的需要。

在这里，需要强调一些 key-list 的概念，在实际场景中有大量需要 key-list 的地方。比如：论坛中的主题列表、回复列表，微博中的用户关注列表、用户 feed 列表、用户关注 feed 列表等等。如果使用 key-value 中的 value 来存储 list（比如：list 打包成 json 放入 value 中），其操作性能是非常低效的。

理想的 Key-list 通常需要如下特点：

1. list 是海量的、且操作性能高效
2. list 是有序的、且可动态调整顺序

至于为什么开源？一方面，我们很多工作都得益于已有的开源系统，所以回馈开源社区是我们应做的义务；另一方面，技术分享也有利于公司本身技术的成长，并吸引更多的技术人才。

InfoQ：能介绍一下 **Memlink** 的特性吗？

冯勇：Memlink 是一个高性能、持久化、分布式的 Key=>List/Queue 数据引擎。正如名称中的 Mem 所示，所有数据都建构在内存中，保证了系统的高性能，同时使用块链进行内存压缩，使用 redo-log 技术保证数据的持久化。此外，Memlink 还支持主从复制、读写分离、数据项过滤操作等功能。

特点：

- 内存数据引擎，性能极为高效
- List 中的 Node 采用块链组织，精简内存，优化查找效率
- Node 数据项可自定义 Mask 表，支持多种过滤操作
- 支持 redo-log，数据持久化，非 Cache 模式
- 分布式，主从同步
- 读写分离，写优先处理。

InfoQ：我们知道市面上还有一些其他基于内存的数据引擎，比如 **Redis** 和 **Scalaris**，跟它们相比 **Memlink** 解决了什么特别的问题吗？

冯勇：在设计和开发 memlink 之前，我们也认真分析对比了 Redis。最终没有采用 Redis 原因有以下四点：

1. Redis 持久化策略(redo-log)不能完全满足线上生产的需求。对于一个成熟的互联网应用应该有足够的容错能力。比如系统统重启、宕机等而不丢失数据。Redis 持久化策略一：定时同步磁盘（此期间重启会丢失部分数据）；持久化策略二：不断追加 log，这样容易使 log 膨胀，性能降低。Memlink 持久化策略是同时借鉴 Redis 两种策略，在非创建快照期间追加 redo-log，在完成快照后清除 redo-log。
2. Redis 主从同步策略不够完善。比如：slaver 因为某原因丢失了部分同步数据，则需要重新完全获取一份主节点的所有数据。在大数据量的情况下，不太合适线上生产的需求。
3. Redis 单线程模式，读写没有分离，只能使用单核。Memlink 为多线程，充分利用多核，并进行了读写分离，优先保证写。
4. 在内存消耗和性能上 Memlink 要优于 Redis。

Memlink 是 key=>list/queue 引擎，Scalaris 是 key-value，两者功能出发点上不一样。

InfoQ：**Memlink** 在天涯内部的哪些系统中得到了采用？可以提供一下 **Memlink** 带来的性能

变化的数据吗？

冯勇：Memlink 主要应用于天涯论坛类型产品(论坛、来吧)中。比如论坛的主题列表，当数据达到百万、千万量级，采用 Mysql 系统进行分页浏览时，基本上不能响应，而 Memlink 则性能提升了上百倍。具体可见 Benchmark。

InfoQ：能向广大的开发者朋友们介绍一下，如何来选择一款适用自己的 **NoSQL** 产品呢？

冯勇：首先需要确定业务需求，是否需要 NoSQL 产品。对于大多数百万量级、千万量级的应用，MySQL 也能支持。

其次在明确需要 NoSQL 产品后，应根据业务需求抽象出数据模型，比如：有些数据是需要采用 key-value 系统存储，有些数据是需要采用 key-list 系统存储，有些数据是采用文档数据库存储等等。

对于 NoSQL 产品候选列表的选项，可以从如下维度进行考虑：

1. 系统的容量、性能、软硬件环境是否符合需求？
2. 数据的安全机制如何？各种异常是否会丢失数据？
3. 具备主从复制功能？何种一致性策略？
4. 可扩展性？自动扩展 or 程序进行扩展？
5. 系统的可控性？系统的成熟度、对开发者的支持度、bug 谁来修复等等

InfoQ：**Memlink** 现在的版本号是多少？未来的发展计划是怎样的？

冯勇：Memlink 现在的版本号为 0.2，具备基本 key-list/主从复制等功能，目前正在测试中。

在 0.3/0.4 版本中，Memlink 会增加双向队列、用户认证等功能。具体可以见 Memlink 的 RoadMap。

长远而言，Memlink 专注为一个高性能、持久化、分布式的 Key=>List/Queue 数据引擎，不会增加其他数据存储模型。

更多关于 Memlink 的信息，请参考 Memlink 的介绍文档和设计文档。

原文链接：http://www.infoq.com/cn/news/2010/11/tianya-memlink

Apache 怒了，威胁说要离开 JCP

作者 Alex Blewitt 译者 张龙

近日，Apache 软件基金会董事做出了一个史无前例的举动：如果他们作为 Java 规范实现者的权利得不到满足，那么就打算离开 JCP。

> 在 JSPA 上，Oracle 与 ASF 同意一起参与 JCP 的建设，ASF 享有 Java SE 测试包（即"TCK"）的许可权，这样 ASF 就可以基于 Apache License 测试并发布 Apache Harmony 项目了。但现在 Oracle 违背了他们所签订的合同契约，只提供了附加额外条款与条件的 TCK 许可，而该许可与开源或免费软件许可又不兼容。
>
> ASF 声称如果他们作为 Java 规范实现者的权利得不到 JCP 执委会最大限度的满足，那么他们将终止与 JCP 的合作关系。缺乏积极、有力、清晰执行力的权利反映出 JSPA 是毫无价值的，只能证明 JCP 规范只不过专有文档而已。

最具争议的事情就是 Sun（现在的 Oracle）拒绝为 Apache 开源的 Java 实现提供测试兼容包（TCK）。其中的原因在于财政—— Oracle 很大一部分收益来自于移动设备上的 Java，而移动设备上免费的 JVM 可能会威胁到 Oracle 的收入来源（OpenJDK 对于很多嵌入式设备都是不适用的，因为它要求所有的源代码都得基于 GPL 发布，这就需要商业许可了）。

然而，JCP 文档还需要使用 JSRs。对于 Java 6 平台的 JSR 270 来说，许可条款包括：

> 10.应该向非营利组织与参与创建 Java SE 6 规范兼容性实现的个人无偿提供 Java SE 6 TCK，只需带有商标与商标版权即可，但不提供支持。

从 2007 年 4 月开始 Apache 董事就开始抱怨这种矛盾，然后又是 Java 5 规范。现在，主要的问题在于使用限制上：

> 从 2006 年 8 月开始，ASF 就开始尝试从 Sun 搞来 Java SE 测试包的许可。该测试包叫做"Java Compatibility Kit"（即"JCK"），Apache Harmony 项目需要使用它说明它与 Java SE 规范的兼容程度，而这正是 Sun 规范许可所要求的。Sun 所提供的 JCK 许可通过限制"

使用条款"对软件用户增加了不少知识产权方面的约束。

这些限制与 Java 规范参与协议（JSPA）之间矛盾重重——JCP 的条款是由 Sun 所统治的。特别地，JSPA 要求：

1. 规范领导不能"增加任何契约来限制任何许可创建或是分发独立实现的权利"（第 5.C.III 节）。
2. 规范领导必须为任何兼容的规范实现提供必要的免知识产权的许可（第 5.B 节）。

该实现正好与 Oracle 和 Google 之间的专利诉讼交 叉，而后者的诉讼主要关注在 JVM 本身。虽然 Harmony 带有 DRLVM（最初由 Intel 捐献），但 TCK 本应该授权使用这些专利。要是没有 TCK，Apache Harmony 的 JVM 可能就会违背某些专利——但 Oracle 发起的庭审自动卷入到了关于 JSPA 的争论，Oracle 希望避免这些情况的发生。

由于 IBM 加入到了 OpenJDK 当中，人们不禁要问：Oracle 能否拯救 Java 呢？Oracle 的立场依旧强硬——不管之前人们怎么说，在最后验收前，应该将这种残局公之于众。

然而遗憾的是，虽然 Oracle 仍旧需要人手，但 Doug Lea 等人却已经离开了 JCP。

无论 Apache 是否还留在 JCP，Oracle 已经不太可能改变他们对无限制 TCK 的立场了。即便 Apache 还留在 JCP 并且对 Java 7 投了反对票，Oracle 也会竭尽所能地推进 Java 7。

Oracle 已经将自己推向了社区的角落。MySQL 之前的众多开发者都选择了离开（并且创建了开源产品，如 MariaDB 与 Drizzle）；OpenSolaris 董事也已离职去了 Illumos，还有其他很多优秀人才也都离开了。人们甚至还发起了一个投票，质疑 JCP 是否还是一个独立的标准化组织。

自从 Oracle 开始控制 Java 与 JCP 起，他们就可以像终止 OpenSolaris 项目一样轻松终止 JCP。由于 Apache 没有足够的资源支撑起这 场战役，因此无论他们是走是留都对明年 Oracle 发布 Java 7 没有太大的影响。结果，离开 JCP 可能是 Apache 唯一的选择。

原文链接： http://www.infoq.com/cn/news/2010/11/apache-jcp

相关内容：

- IBM加入OpenJDK，和Oracle一道推动Java发展
- JCP执行委员会新成员选举结果揭晓：Hologic未获通过
- 与Josh Bloch探讨Java未来
- Bundle.update：Java EE中的OSGi、JSR 294 被叫停、JSR 310 Java Date与Time API

热点新闻 | News

讨论 :研发团队中最合适的开发测试比是多少？

作者 霍泰稳

在上周六淘宝网和阿里巴巴 B2B 联合主办、InfoQ 独家社区支持的第二届互联网测试交流大会上，来自 Google、Baidu、网易、腾讯、淘宝、阿里巴巴、FreeWheel 等公司的测试经理分别分享了他们在测试领域的心得体会，原计划 500 人的会场到会了 800 人。在讲师交流环节，大家对“研发团队中最合适的开发测试比是多少”话题做了讨论。（编辑注：方便起见，本文对各测试经理均使用花名，请勿对号入座！）

郭靖是某大型网上商城测试团队的负责人，自 2003 年加入公司后就组建了该网站的测试团队，推动团队进行性能测试、自动化测试、接口测试等。随着团队的壮大，他遇到的比较头疼的问题是一个研发团队中开发测试比究竟多少比较合适？根据他的了解，Google 的开发测试比是 10 比 1，微软的比例是 1 比 3，大家都做得很好。在长时间的纠结之后，他问自己，这个比例真的很重要吗？其实在他看来，大家关注的是开发测试人员的技能、效率、成本和质量等维度，而不是某个具体的比例。同时，他也疑问能否有一种合适的手段来说清楚某个研发团队最合适的开发测试比是多少。

来自另一家知名互联网公司的测试经理，也是负责整个公司测试框架的杨康谈到自己对这个问题的理解：

> “公司高层通常会给质量部定一些目标，要求你的人员配比是多少。但我的理解是，很难定义和衡量质量部的工作成果。对于高层来说，给质量部最有力的一个施压办法就是人员配比，至于下一步你怎么去做，你自己去想办法。坦白来说，高层是希望能有一个手段，让质量部的人员能尽可能快地完成工作。在开始的时候，我们整个研发团队也有过 1 比 2 或者 1 比 3 的时候，高层就不断地要求我们降低开发测试人员比。但是现在大家已经都不提这个事情了，因为高层认为我们现在做的还可以，对产品的贡献还不错，现在提的更多是如何更好地提高质量。

托雷同样来自一家大型互联网公司，也是测试团队的负责人，就杨康提到的和公司高层相关的问题，他同样深有感触：

> 在我加入公司后，我的直接老板换过几次。每次新的老板上任，都会问这些问题，我都需要重新给他讲一遍我们是如何做测试的，开发测试比是多少等等。做了一两年之后，他基本就不问了，但是不幸地是老板又换了，第一件事又是来问这个问题。然后自己就将 3 年前类似的话再说一遍。我想大家都了解这种情况，有时主管测试的高层也许是测试领域外的，可能很多是开发出身，不一定真得懂测试。所以我就想能否找到一种比较科学的方法，像性能测试一样，能大体看到它的瓶颈，看到它的容量规划到什么样的程度等。

来自新创互联网公司的测试经理欧阳克从“存在即合理”的哲学角度分享了自己的观点：

> 开发测试比其实反映的是公司内部开发和测试结构的关系，有的公司开发人员多一些，测试人员少一些，这个比例相应来说是最低的。而有的公司比较依赖于 QA，可能开发测试比就高一些。所以不同的公司，差异肯定会比较大。可能没有一种合适的办法能数清楚这个事情。我的看法就是存在就是合理。

对 Google 研发团队比较有了解的洪七公介绍了 Google 开发测试比 10 比 1 的背景，同样认为开发测试比不是最重要的，最重要的是要根据公司的情况确定合适的比例：

> 我非常同意郭靖的观点，其实高层经常的问题就是人员配比，因为这个问题大家都理解，而像缺陷比率这些问题，有些人可能根本就不懂。而且人员配比和成本非常相关，老板都关心这个问题。所以说和高层沟通的时候，一定不能纠缠于这个问题，你要告诉他人员比例不是最关键的问题。如果他非要有个标准逼着你这样去做的话，你要给他讲其他公司的人员配比的背景。
>
> 比如 Google 的开发测试比是 10 比 1，其实是有前提条件的：第一，测试部门并不是要进入所有的项目:，如果你的项目做得太糟，程序写得太烂，测试人员压根就不会进去；第二，测试人员有绝对的权力控制自己的资源；第三是开发工程师自己要做大量的测试。如果没有这三个条件，你怎么去谈 10 比 1？根本是不可能的事情。所以你一定要告诉老板，开发测试比不是最重要的，我们要根据公司的具体情况确定合适的比例。

究竟某个研发团队的开发测试比多少合适，这和团队的开发、测试工程师的素质也有密不可分的关系，一灯大师从大家对测试的偏见，结合 Google 对开发和测试的理解，解释说要压缩开发测试比，可行的办法是提高测试工程师的素质：

> 10 比 1 只是一个量的问题，其实背后还有一个质的问题，即人员素质的问题。在很多传统软件公司中，很多测试人员的素质不是非常高，比如大家通常说的观点就是“你做不了开发就去做测试”。像这种情况必须要扭转。在我们公司，现在就特别重视提高测试人员的技能、素质，同时也一直在提高招聘门槛，加强对测试人员的培养，包括调教开发人员配合做单元测试等。所以说 Google 的研发比例是 10 比 1，这里面不能忽视的一点是 Google 的研发人员和测试人员素质都很高，尤其是研发工程师对单元测试和代码质量要求非常高。而咱们其他的一些互联网行业的研发、测试人员可能就没有这么高的素质，这种情况下只能采取人海战术。

读者朋友，您的公司的开发测试比是多少？您对这一话题的看法是什么？欢迎跟帖讨论。浏览更多 InfoQ 中文站的测试内容，请点击测试专题。

原文链接：http://www.infoq.com/cn/news/2010/11/discussion-dev-tester-rate

相关内容：

- 颠覆者生存：互联网产品测试观点
- 讨论：测试用例的粒度——粗细之争
- 百度技术沙龙第 4 期回顾：Web测试自动化（含资料下载）
- Web自动化测试工具Selenium 1.0 正式发布:Chrome支持+用户指南
- 为Java和Flex编写Mock对象

iPhone 的开发环境优于 Android 吗？

作者 Jean-Jacques Dubray 译者 池建强

随着移动应用的日益普及，很多人开始尝试比较不同开发环境下的开发体验。大约一年前，David Green 发表了一篇针对 Android 和 iPhone 开发环境的比较文章，而 John Blanco 在上周也针对二者进行了比较分析。Dave 和 John 都同意：

> 使用 Java 好于 Objective-C。私有方法、内部类、匿名类、泛型、更好的功能语法和广泛的第三方代码仅仅体现了 Java 优势的一小部分，这已经不是竞争了。

John 和 Dave 在 Xcode 与 Eclipse 的比较上，意见就没那么一致了：

> [John] 我曾经喜欢过 Eclipse。我能掌握一个 IDE 并在工作中获益。从我不得不使用 Eclipse 以来已经一年多了[...] 收获的是可怕的使用体验...我不知道是怎么发生的。Eclipse 臃肿，速度慢，改变编辑器上下文环境（XML vs. Java vs. Android Manifest 等）这样的简单行为都会让人心烦意乱，这些动作居然需要几秒钟的时间。[...]它使得做 Android 开发变成了“悲惨”的经历。与 XCode 相比，XCode 能为你带来快乐的工作。它优雅、快如闪电，在写代码时我从未看见 XCode 发生迟缓的现象。我对 XCode 给与肯定。 Xcode 占有压倒性优势。

Dave 认为 Eclipse 的增量编译器是其主要优点。对他来说内容辅助做的并不好，而 XCode 则打开了太多窗口：

> XCode 是如此糟糕，我甚至不知道从哪开始。我觉的 XCode 要想成为可行的 IDE，至少应该完善我提到的最小列表中的特征。

不过 Dave 喜欢 iPhone 的 Interface Builder。

从模拟器的角度来看，John 觉得 iPhone 的模拟器在真正模拟 GPS 或加速器方面还有一些欠缺。不过 iPhone 模拟器的准确性给他留下了深刻影响：

> 我很少见到在设备上出的问题在模拟器不出现的情况。iPhone 模拟器运行很快，我随时可以关闭它，可以很容易进行复位，改变语言等。相反，Android 的模拟器是我见过最烂的模拟器。比黑莓的还烂——这说明了一些问题。有时我在模拟器上运行一个程序，无论如何都不能运行，我不得不重新启动。另外它还慢的要命。有一次我想显示某个功能居然用了 8 秒钟。这正好和我提到的 Eclipse 问题相关，之所以指出这些问题，是因为我在 iPhone 开发环境中没有遇到任何以上问题。我碰到的每个 Android 开发者都说，他们几乎不怎么用模拟器，而是直接在设备上运行程序。

Dave 看到的是 Java 开发环境十年不懈改进的能力，包括全功能调试器、使用 JUnit 的能力和内存配置工具。

20 年前的秋天，我写下了第一行 Objective-C 程序。我记得那真是个神奇的经历，以至于我不能长时间离开宾夕法尼亚州立大学的计算机实验室。对于很多人来说很难想象，在那样一个遥远的过去（NeXTStep），XCode 会和今天一样犀利。Interface Builder 作为 NeXTStep 的基础功能，是 Jean-Marie Hullot 的智慧结晶，他从 1985 年就开始做这方面的工作，NeXTStep 引导 Tim Berners-Lee 在 CERN 建立了 Web 的基础。与他一起的还有 Jack Greenfield，他开发了第一个 ORM 工具， 为模型驱动引擎建立了坚实的基础。有了这样一个创新的历史，也就难怪 XCode 依然可以运行在 2010 年了。但是它现在仍然与时俱进吗？还在领先么？能持 续改进吗？与其他语言相比，Objective-C 还是一门足够先进的语言吗？或者，XCode 已经落后了，会因为更先进的、模型驱动工具的出现而不再流 行？我们甚至可以问 MVC 是未来吗？或者，Android 的概念和方法更优越？

对此你有什么考虑？

原文链接：http://www.infoq.com/cn/news/2010/11/iphone-android-dev-env

相关内容：

- Adobe AIR登陆Android平台
- 苹果放宽iPhone开发工具和数据共享限制
- 本地（手机）应用未来会怎样？
- MonoTouch：用.NET开发iPhone应用
- 开发者眼中的Android手机平台

热点新闻 | News

HTML5 vs. Flash：Adobe 面向何方？

作者 Dionysios G. Synodinos 译者 杨晨

看来去年 Adobe 就要转向 HTML5 并且将其整合入产品战略中。在本次 Adobe 开发者大会（MAX）上，演讲者 Lee Brimelow 尝试向开发者阐述公司的 HTML5 战略，尤其是 HTML5 在桌面以及移动平台上和 Flash 以及 Flex 的竞争。

Adobe Illustrator，Adobe 的旗舰级产品之一，最近也拥有了 HTML5 扩展包。它提供了对 HTML5 和 CSS3 的基本支持，并且扩展了 SVG 能力以及可以协助开发者设计 web 和设备内容：

Illustrator 的 HTML5 扩展包提供的特性能够帮助用户利用 HTML5、CSS3、SVG 和 Canvas 的最新特性来支持多屏幕多分辨率开发。SVG 和 Canvas 的强化同时也简化了交互式网页的开发。在数据驱动的 web 工作流中，用户也同样可以通过生成 dynamic vector art 来使用 Illustrator 创建 web widgets。并且可以将数据从 Illustrator Appearance Panel 导出到 CSS3，在不同的开发者之间交换。

同时 Dreamweaver CS5 也能够支持 HTML5，最开始只是通过插件支持，升级到 11.0.3 之后开始提供原生支持。这次升级能够原生支持新的 HTML5 结构化元素以及最新浏览器支持的 CSS3 属性。众多新特性的其中一个便是多屏预览面板，这个特性能够显示三种不同的分辨率下的页面效果，例如桌面显示器，平板设备和移动手机。它允许你使用 CSS3 media query 在不同的设备上使用不同的样式表，这样页面在不同平台上的渲染效果都能够令人满意。

InfoQ 曾经报道过，在 Adobe MAX 上，Adobe 发布了一个 HTML5 动画效果的 IDE 预览版。这个开发代号叫 Edge 的 IDE 使用的是 WebKit 渲染引擎来预览动画效果，也能够提供和 Dreamweaver 类似的源码编辑功能。Edge 的 UI 和 其他 Adobe 工具的风格类似，但是 Flash 开发者会更加熟悉一些。在内部结构中，Edge 使用 jQuery 和 CSS 来管理 demo 内部渐变的过渡，貌 似也是同样使用了 jQuery UI 和 Easing 插件。

不仅如此，在 Adobe MAX 上，平台的宣传者 Lee Brimelow 还做了时长为一个小时的讲演，揭示了大量有用的信息，例如 Adobe 对 HTML5 的看法以及和 Flash、Flex 的关系。 他说，当年 iPhone 没有 Flash 支持，这并不是大问题，他能够理解关于这个决定的一些不同意见，

但是 iPad 没有 Flash 支持就相当令人失望了。 而且我们也意识到了“这有足够的动力来完成这个”。Lee 也代表他自己表达了对于 iPhone 新的许可证协议的失望，它并不允许第三方开发工具。这也事实 上宣布了 Adobe 的 CS5 套件是不可能将 Flash 引入 iPhone 了。在他的演讲中，Lee 也提到了“HTML5 并不是 Flash 的竞争者或者替代 品。你可以在浏览器中完成标准制定的每一件事情，但是如果你需要做得更多，那么这就是需要 Flash 的时候了”。

虽然 Lee 说“HTML 是 Web 的基础，你不可能违逆”，但是现在仍然太早下结论，因为“60% 的用户仍然不能查看 HTML5 内容”。他接着提供了一些论点，指出从浏览器市场占用率的角度来看，HTML5 仍然没有被广泛接受。

在列举了 HTML5 的一些经常被错误引用的信息之后，Lee 给出了他自己的理解，他强调 Adobe 将大量精力投入到 HTML5 工具上，并且敦促开发者开始学习 HTML5，尤其是 Flash 开发者。至于视频播放，现在仍然是 Flash 的强项，但是 Lee 建议如果你需要获得最广泛的观众群，那么最好能够同时支持 Flash 和 HTMl5 视频播放。他没有特意提到 iPhone 和 iPad，但是其他的移动设备平台也缺乏 Flash 支持。话虽如此，他重申了他坚信 “Flash 能够提供更好的视频”，不仅仅在特性支持方面，还包括视频质量，所以它不应该只是一个备选方案，而应该是首选。

Lee 最后解释了 Adobe 的对移动 Web 和 Flash 应该如何成为移动平台的一部分的观点。虽然 Flash 应该被小心使用，因为它并不是普适的，而且现存 Flash 桌面内容可能不适合移动设备。他也建议对特定应用来说，AIR 是正确的选择，而且根据你选择的特性，很多时候你将能够得到原生的应用。

这仅仅只是 Lee 的一家之言，通过 Adobe 最近的动作和 Lee 在 Adobe MAX 上的讲话来看，我们应该能够期望 Adobe 将会更深入地进入到 HTML5 中，而且会在产品线中提供更广泛的支持。看起来 Adobe 希望 Flash 成为 HTML5 的一个补充产品，而且将精力放在视频和移动平台上。在这些应用场景下，Flash 已经被证明了它要比 HTML5 的产品要优秀得多。

你可以在 InfoQ 上找到更多关于 HTML 5 的信息，或者参加 Qcon London 2010 大会，我们将会有一个完整的 HTML5 平台 track。

原文链接：http://www.infoq.com/cn/news/2010/11/html5-flash-flex-adobe

相关内容：Adobe 发布 64 位 Flash Player 预览版，支持 IE9、Flash vs. HTML5：RIA 领域当前的混战、跨平台部署和企业应用：Adobe Flash Platform 最新进展、Adobe Flash 平台开发者峰会谈 RIA 开发现状、诺贝尔奖官网——纯 Adobe 应用

Adobe Flash Builder 4 简体中文正式版

Adobe® Flash® Builder™ 4（之前称为 Adobe Flex® Builder™）软件是一款专业级的开发工具，可以帮助软件开发人员使用开放源 Flex 框架快速开发跨平台富 Internet 应用程序 (RIA) 和内容。。Flash Builder 4 有许多新的特性，可以与 Adobe Flash Catalyst 集成、以数据为中心的开发、与 Adobe Creative Suite 设计工具集成、交互式数据可视化、智能代码编辑和调试等。

现在就来下载使用

即可体验流畅的开发过程，集成更多基础设施

开发更具表现力的应用程序

Adobe Flash Builder 4 简体中文正式版285M

立即下载

InfoQ中文站提供以下Adobe Flash Platform相关产品和工具的高速下载：

- Flash Platform一览表
- Flash Player
- Flash Catalyst Beta 2 292M 立即下载

特别专题 策划：InfoQ 中文站； 执行：霍泰稳 葛明见

随着内容和形式的不断发展改进，HTML5 的规范标准已经渐渐被大家所接受，但是目前关于 HTML5 的争议还是特别的多，有的已经开始在网站上尝试应用，有的却认为 HTML5 现在还不适宜大规模推广，还需要完善。本期 HTML5 特意邀请和采访了相关技术领域的专家，为大家展示他们的观点。

HTML5 时代开端

当 HTML5 来敲门 — Opera 技术布道专家谢子斌谈 HTML5

HTML5 在豆瓣中的应用 — 使用 HTML5 构建下一代的 Web Form

特别专题

当 HTML5 来敲门

作者 陈达

偶然一次拿起 Amazon Kindle DX ,面对着大屏幕 ,习惯性地用手指划动屏幕试图翻页失败后，猛然发现自己的操作习惯已经被改变。对着大屏幕的掌上设备，已经习惯了触摸，习惯了这种便利。让我想起了"技术驱动生活，技术改变生活"这句话。这是一个最平常不过的技术改变生活的例子。技术有很多种，改变生活的例子也数不胜数。而今天我将以一个前端开发工程师（以下简称：前端）的角度，说说 HTML5 给我们带来的改变和使用 HTML5 所需要注意的点。

现状

从目前的状况来说，HTML5 很流行。Google 在推，Microsoft 在推，很多大公司已经尝试在用，但依然没有大规模的商业应用。原因大概可以归结为两点，一是 HTML5 本身对于技术人员来说还比较新；一是占据市场最大份额的 IE（IE9 Beta 已经支持了相当多的 HTML5 技术）对大部分 HTML5 特性并不支持，其他浏览器的支持也并不全面。

当 HTML5 来敲门

现今的互联网，大部分信息与服务都不再是一家独有，当大家都能提供相同的服务时，体验更好的将会被选择。用户选择了你，信任了你，才可能养活你。所以很多时候，企业要做的是，提升自己产品的体验，留住用户，让他们离不开，让他们掏了腰包依然觉得物有所值。当然，如果能让用户感受到物超所值就更好了。

企业期望着自己的产品更具竞争力；而作为技术人员，我们希望能把产品做得更完美。在拼用户体验的关键时刻，HTML5 应景而生，敲响了我们的大门。它从性能、速度和操作的交互模式上，都给我们带来了全新的体验。我们想利用它。

对于 HTML5 的好，我相信你已经从很多地方看到。这里，我的重点会放在如何应对这个新事物上。可能讲的大部分都是前端应该做些什么，如何去做的事，但这也是在对所有希望提

升自己产品的人说的。

（一）面对新的技术

从某种程度上来说，HTML5 并不是完全新生的技术。而是一个版本的升级，只是这次升级给我们带来了更多的东西。虽然这些东西大部分都还是一份草案，但浏览器厂商非常看重，都纷纷在竞争着去实现。而无论它是否草案，浏览器的支持，给了我们去做更好产品的机会。那么，去尝试吧。而作为前端，我们需要做的，能做到的有哪些呢？

- 技术的学习

对于新的标签的写法，按照是 HTML4 和 XHTML1 的习惯，我们已经可以猜到怎么写。而具体的使用，根据 HTML5 设计原理所遵循的伯斯塔尔法则（Postel's Law），"发送时要保守，接收时要开放"，我们可以很好地理解。比如 <video /> 标签，设计时严格限定 <video /> 标签才是视频播放的标签，但接收的时候，可以有多种格式的 source:

```
<video>
   <source src="movie.mp4">
   <source src="movie.ogv">
   <object data="movie.swf">
      <a href="movie.mp4">download</a>
   </object>
</video>
```

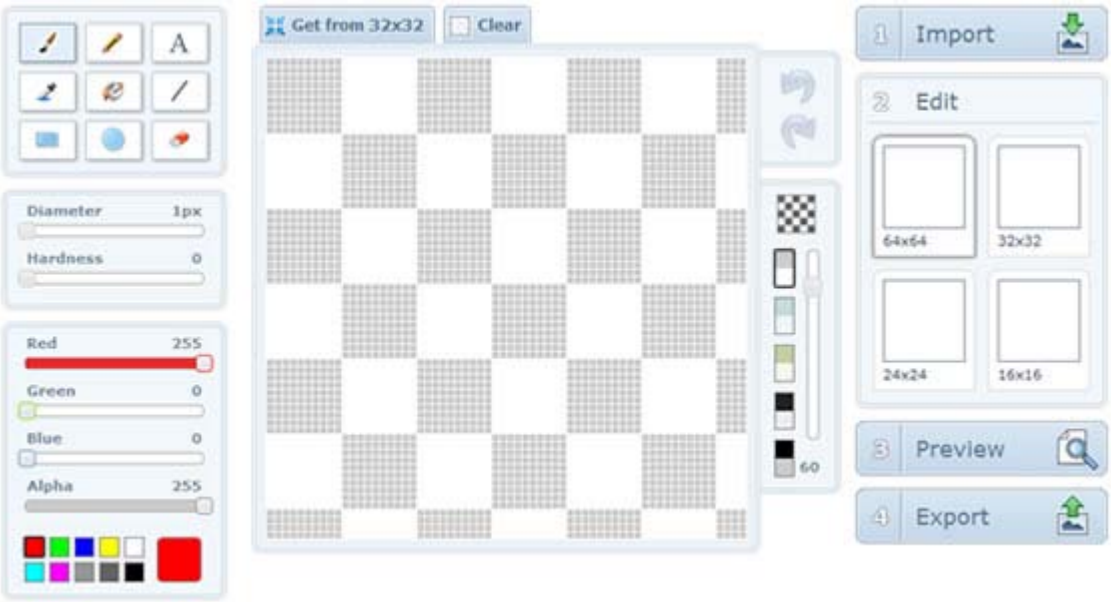
上文所述的改变，要学习，对我们的挑战并不大。HTML5 提供了各种新的 API、新特性，才才是学习的难点所在。比如 Canvas，要做好一个好的 Canvas 3D 应用，不止需要有编程的能力，还需要有图形学方面的知识。又比如 Web SQL Database，还需要前端学会处理数据库。当然，HTML5 的技术远远不止这些，并且很多新技术都还没有人给出相应应用的实现方式，以及最佳实践。学习起来还是有一定难度的。另外，HTML5 在一些浏览器是完全不支持的，比如 IE6、IE7（当然，拖放--Drag and Drop 微软本来就有了），这时要去做所有的折衷方案可能比较难，所以，如何去权衡优雅降级也是一个难题。但，想把一个产品做得完美，或者只是想把一件事做得完美，又有什么事是容易的呢？

- 技术的使用

（1）创新才能让技术真正改变我们的生活

我们面临的最大挑战应该是创新。他不像已有的很多技术，已经有很多实例给我们看，甚至把源代码都开放出来。这就需要我们去了解用户的需求，大胆去为用户想，创造出更便利的体验。比如 X-icon Editor：http://www.xiconeditor.com/，利用 Canvas 实现了一个与桌面程

序一样的图片编辑器：



又如 Minus: http://min.us 利用 File API 和 Drag and Drop 特性，实现了拖放上传的新体验：

这些创新。虽然在桌面软件上不算新鲜，但在 Web Application 上，都是新东西，是需要我们去实现的。而在最初，最直接的"创新"方式可能我们会是把更多桌面应用程序的好体验搬到 Web 上来。其实，这是非常不错的一种方案。因为很多人都习惯了桌面程序，如果 Web 提供了相应的功能，对于操作的方法，他们很快就会上手。这对于用户来说，减少了学习成本。轻松上手也是良好体验的一种。

（**2**）安全是我们应该时刻记住的

记得有一次我在使用 Google Docs 写一篇文档的时候，浏览器突然崩溃了。当时，就习惯性地想，如果 Google Docs 支持 LocalStorage（本地存储） 的话，那么，我下次打开，内容一定不会丢失。但仔细想想：

1. 这是一份公司的内部分档，不希望被人外人看到
2. LocalStorage 会存储在用户使用的电脑上

那么这时，一定不适合用 LocalStorage 这种技术，而应该用 Ajax 自动保存。因为这本东西是内部（对外保密）的文档，如果用户是在公共电脑上使用，可能会造成涉密。相应的，像我们使用 Web SQL Database 这种客户端的数据储存，就更需要有安全意识了。哪些是可以用的，哪些是不可以用的，应该要做好规划和检查。不要让简单的 localStorage.setItem 暴露了用户的隐私。

当然，可能会有人说这是一种设计的不严谨。我想，这种出发点是好的。技术本身本来就是中立的，它只要提供更多方便、灵活的特性给我们，而具体如何去正确运用，应该是作为程序员应该去思考的。

（**3**）规范需要统一

Jeremy Keith 在 Fronteers 2010 上说的 HTML5 设计原理告诉我们，HTML5 最大的一个特点是: 兼容。比如 <input type=alipay> 和 <input type="alipay"> 这样的标签的合格的，写属性的时候可以不用引号， 标签可以自闭合也可以不闭合。而在结构上，甚至可以写这样的标签：

```
<a href="/path/to/resource">
  <h2>Headline text</h2>
  <p>Paragraph text.</p>
</a>
```

这对于写惯了 XHTML 的我们，可能很难接受。但程序这样设计了，它也给我们带来了更多的灵活，虽然与我们已有的风格有所冲突。我们当然希望可以按照这些灵感，用自己的风格去编码，但团队合作最忌的就是没有统一的风格，这会阻碍我们前进的步伐。因此，面对 HTML 5，我们需要一套规范，来让我们的团队更好地进行协作，而不是阻碍协作。

（二）面对其他部门的人员

作为对 HTML5 最了解的一群人。前端应该负责去推动这样的技术应用。当然，并不是说只有前端才希望产品有更好的体验，而更多的是因为，只我们了解。想要应用新的技术，创建新的体验。只有执行的这群人了解了相应的，才有可能用上。对于交互设计师、视觉设计师

和后端（服务端）工程师，还需要我们去告诉他们。而这也是我们的职责所在。

比如，你要告诉交互设计师，现在 Web 上 Drag and Drop 的支持比较好了，可以设计一些这方面的交互方式；你要去告诉视觉设计师，Drag and Drop 的事件过程，让他们像知道 a:hover 一样，会给你设计出鼠标经过时的样式；你还要去告诉后端工程师，其实 cookie 有时候已经用不了上，因为在 IE 我们可以用 userData，而其他的可以用 LocalStorage 来实现，这样可以不在 Header 中传送，页面将会更快。

需要我们前端去做的，除了去学习，去做，就是去告诉他们有哪些新东西，甚至让他们也学习一点这方面的知识。只有他们了解了这些，将产品设计和前后端技术配合起来，对产品才可能有质的提升。

总结

其实这一切，所有的新技术。都是为了更美好的生活而做的。HTML5 也是。或许某天我们再也不需要写像引用 flash 这样晦涩的代码了，而只要写像 <video src="movie.mp4" /> 这样简单的标签。

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
   codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflas
h.cab#version=6,0,40,0" WIDTH="100%" HEIGHT="100%" id="someflash">
   <param name="movie" value="someflash.swf" />
   <param name="quality" value="high" />
   <param name="wmode" value="transparent" />
   <embed src="someflash.swf" quality="high" wmode="transparent" width="100%"
      height="100%" name="someflash" align=""
      type="application/x-shockwave-flash"
      luginspage="http://www.macromedia.com/go/getflashplayer"></embed>
</object>
```

所以，我们应该以更开放的心态去接纳。像伯斯塔尔法则（Postel's Law）所说的：发送时要保守，接收时要开放。当 HTML5 来敲门，我们应该用更开放的心去接纳，用更严谨的态度来编码。

关于作者

林建锋（sofish），目前供职于支付宝，前端架构组成员。主要负责支付宝样式库的构建，致力于样式兼容研究、新技术研究，以及样式框架的开发。个人网站：http://sofish.de。

另外，感谢淘宝网小马（赵泽欣）、支付宝么么茶（吴振昊）对 InfoQ 中文站的大力支持。

特别专题

Opera 技术布道专家谢子斌谈 HTML5

作者 谢子斌

随着 HTML 5 技术的不断发展，有关它的话题讨论越来越热烈，比如 HTML 5 存在哪些不足、开发人员应该尝试还是旁观等，InfoQ 中文站就相关问题专访了 Opera 技术布道专家谢子斌。

InfoQ：请您先介绍一下 **HTML 5** 的历史。

子斌：众所周知，HTML5 是 W3C 的一个规范。如果追溯起源，并非 W3C 提出的 HTML5。它最初起源于一个叫做 WHATWG 的小组。2004 年，W3C 正建立 XHTML2 的规范，Opera 和火狐当时给 W3C 提供了一些改善 HTML 标准的建议，但是 W3C 拒绝了这个报告。Opera 和火狐还有苹果三个公司的技术人员共同成立了 WHATWG 小组，开始一起制定 HTML4 的规范，但还是不叫 HTML5。直到 2007 年 W3C 决定开始制定后 HTML4 的规范，我们才把自己写的一些规范并入 W3C 刚成立的 HTML5 小组。

InfoQ： 浏览器厂商对 **HTML 5** 的发展持怎样的态度？有哪些举措？

子斌：W3C 有 700 多个会员，其中包括很多浏览器厂商，谷歌、微软、Opera、火狐等。他们参与了规范的制定，在推出之前，争执和讨论早已发生，内部已经统一了对规范的看法，所以对 HTML5 的支持，大家基本都是一致的。虽然有我们不满意的地方，但是我们在实施 HTML5 的时候基本是希望看到兼容性的，比方说你是第一个实现 HTML5 某个技术的浏览器，他肯定也希望其他浏览器厂商能实现这个技术。因为只有大家都实现这个技术的时候，程序员才会采用这种技术。如果只在特定浏览器才能实现，那么程序员可能就会采用这个技术应用。这种技术上的实现和推广可以比作一个长跑，可能是你暂时领先，也可能暂时是我领先，但是只有大家都跑到终点，这场长跑才会结束。

InfoQ：对于 **Web** 应用的架构师和开发人员来说，现在应该如何对待 **HTML 5**？尝试还是观望？

子斌：每次和程序员交流的时候，他们都会有这个疑问。我觉得大家使用 HTML5 只是一个时间问题。不管你现在使用，还是两年后再使用，但是最后肯定是要使用 HTML5。我们在推广的时候，就考虑到这个方面，采用所谓优雅降级的做法，可以尝试提供一些 HTML5 技

术给前卫的用户来体验，同时也不拒绝一般用户，比如说 IE6 的用户，虽然不能使用 HTML5，但是可以浏览加入该技术应用的网站。比方说 CSS 3，它不是 HTML 5，但有些人把他们混为一谈，你用它在图片上实现一个圆角的效果，我觉得可以采用这种优雅的方式，如果你的浏览器可以实现，你就给它一个圆角的效果，如果不能，则给它一个一般的效果就可以了。大家都可以浏览你的网站。再比方说 video 标签，IE 6 虽然不能使用这个标签，但是它能读取里面的内容。

InfoQ:据您了解，目前 **HTML 5** 在企业级应用中的使用程度如何？有哪些成功案例？

子斌：有挺多的网站都使用 HTML5 了，比方说 youtube，他们就开始使用 HTML5 的 video 标签了。之前 youtube 上面都是 flash，他们现在也有一个 HTML5 版本了，目前应该正在测试中。

InfoQ：**HTML 5** 存在哪些不足？

子斌：每个语法都有他的缺点，HTML5 当然也有，也许有人会说 HTML 5 若干年后才能应用，媒体也有这样的报道出现。在这里就可以提到 HTML5 的规范上面来，因为这是一个多人参与的透明的过程，如果进度快的话，可能就会有很多缺陷产生。所以这不可能是一次性完成的事情。因为规范太多，只是 HTML5 的规范目前就有 900 多页，然后再加上与 HTML 有关的 Canvas、SVG、离线储存等等，如果想要很好的把它们和 HTML5 统一起来，不可能是短时间就能完成的事情。不过话说回来，需要一点时间完善这些规范也不算是一个缺点。

另外的缺点的就是，在中国，可能有不少用户使用 IE6，在 IE6 上 HTML5 是不兼容的，这个问题比较棘手，但是微软已经意识到这个问题，所以在 IE9 的开发就比较关注 HTML5。当然肯定还有其他缺点，虽然目前还是有很多，但是我觉得优点远远多于缺点，而且完备是需要一个过程的。

InfoQ：最近，有关 **HTML 5**、**Flash** 和 **SilverLight** 之间的讨论很多，您有何评价？

子斌：我觉得 Flash 技术做的非常好，他让开发人员可以使用很多之前不能用的技术。当然 HTML5 一些技术和 Flash 相似的地方，Canvas 和 SVG 相比 Flash 的优势源于 W3C 里面多家浏览器厂商参与技术的透明性，Flash 技术毕竟是一家公司掌控。

Silverlight 和 Flash 情况是差不多的，始终由一家公司控制，他就像一个黑箱，不透明。如果收费的话，大家只能给钱。而 HTML5 则是透明开放的，因为几百家公司都有参与，是免费的。其实 Flash 之所以受欢迎，主要 Adobe 公司提供了一个很好的 IDE 界面。HTML5 缺乏一个强大的 IDE。

InfoQ：您在推广 **HTML 5** 技术的时候做了哪些努力？

子斌：HTML5 必将成为主流，我们最想把 HTML5 里面的规范最大限度的推广给用户使用。规范写好，给浏览器厂商实践，然后推给程序员使用。我们最近也刚刚成立了 HTML5 中文小组，由我负责组织这个小组，目的在于让使用中文的人参与规范制定，比方说在 CSS3 里有个网络字型的技术，打破了本地电脑的局限，但是对于中文来说，会有这样一个奇怪的现象，因为它的字体繁多，即使可以浏览，但下载需要花很多时间，W3C 以前可能没有注意到这一方面，现在成立这个小组就可以提出使用中文的人的一些看法。

InfoQ：感谢您接受采访！

HTML 5 中文兴趣小组章程：

HYPERLINK http://www.w3.org/html/ig/zh/charter.zh.htmlHTML5"
http://www.w3.org/html/ig/zh/charter.zh.htmlHTML5

HTML 5 中文兴趣小组讨论：

HYPERLINK "http://lists.w3.org/Archives/Public/public-html-ig-zh/"
http://lists.w3.org/Archives/Public/public-html-ig-zh/

作者简介

谢子斌 Zi Bin Cheah，马来西亚人。目前就职于 Opera 挪威，负责互联网标准的推广，经常在世界各地进行 HTML5、CSS3 等最新技术的宣讲，他同时也是 W3C HTML5 中文小组主席，HTML5 小组成员。

另外，感谢盛大创新院郭忠祥、周裕波先生对本次采访的支持！

特别专题

HTML5 在豆瓣中的应用

作者 石岩

作为处于发展阶段的新一代 Web 开发标准，HTML5 正以其简洁的书写规范、良好的兼容性以及务实的原则为越来越多的网站所接受，这是 WHATWG 与 W3C 共同努力的结果，作为标准的参与制订者，各浏览器厂商也在积极地为其提供支持。豆瓣一向热衷于新技术的探索和创新，前端团队在 HTML5 与 CSS3 方面也有过很多尝试，从视觉效果到富媒体内容再到本地存储等等，其中一部分已经应用于线上产品，还有一部分以浏览器扩展、客户端、手持设备应用等形式服务于不同需求的用户。下面我就来简单介绍下：

在线上产品中的应用

豆瓣目前主要的流量来源依然是 PC 端的 IE6，这是个颇具中国特色的无奈事实，但我们同时发现，豆瓣的 Chrome 和 Firefox 用户量在不断增长，特别是 Chrome，目前已超过 Firefox 位居第二，这让我们有了充足的理由在产品中加入 HTML5 特性来提升用户体验。

劝导用户升级浏览器是个需要耐心的工作，一般用户并不会因只停留在单方游说层面的所谓更快、更安全、更符合标准做出任何反应，毕竟改变习惯是个痛苦的决定，引导方式不正确会让用户烦躁，甚至对标准浏览器反目成仇。那何不换个思路，先为一部分勇于尝试新鲜事物的用户提供更为友好的视觉、交互体验，在他们享用的同时自然会对身边使用过时浏览器的朋友产生一种更为直观的影响。这虽然没有游戏玩家在比对速度与 3D 特效后产生为电脑更新换代的念头来得迅猛，但毕竟是种积极而不唐突的方式。

豆瓣确实这么做了。细心留意可以发现，在近期的产品 UI 中，圆角、阴影、背景渐变均由 CSS3 来完成，这也是视觉设计师与前端工程师加深交流的意义所在，使用 CSS3 来绘制界面节省了大量代码与图片，从而降低了工作量，也为公司节省了带宽，设计师在设计的同时也考虑了在无法识别 CSS3 的浏览器下元素对应的样式，让低端浏览器用户感受到另一种简约的风格。

表单验证方面，我们优先判断 Form 2.0 中新增元素的默认属性，比如 type、placeholder、

pattern、required 等，脚本中对于类型、验证规则的处理只在不支持这些属性的浏览器中生效，让高级浏览器用户体会到原生的执行效率。在读书、电影条目方面我们也开始加入 Microdata 来实现强大的语义结构，让豆瓣不仅能够提供数据，还能提供数据定义。

HSL 是一个全新的色彩空间，它可以方便地定义色彩的色相、亮度、饱和度，这种定义方式对称于亮与暗，更类似于人类感觉颜色的方式。豆瓣小站的风格是用户自定义的，比如导航栏标签的背景色，但标签悬停时的颜色是由系统生成的，生成的规则便是增加当前背景色的饱和度，使其颜色比默认状态下更深，这在传统 RGB 色彩空间中是不方便做到的。

除此之外，豆瓣还在桌面客户端、浏览器扩展、手机应用等方面大量使用了 HTML5 与 CSS3 特性，相较于传统开发方式，这种利用前端技术实现的跨平台跨设备的开发思路有其明显的优势：学习成本低（HTML & CSS & JS）、开发周期短（类似于网站前端开发）、代码复用性强（除了针对不同设备的接口调用外，界面、逻辑部分大多可以复用），这也得益于浏览器厂商、手机厂商对于 HTML5 的充分肯定与积极支持。当然，Webkit 这个优质的浏览器引擎功不可没。

桌面客户端

OneRing 是由豆瓣首席架构师洪强宁开发的一款跨平台桌面应用框架，它支持使用前端技术开发桌面客户端。即将发布的豆瓣电台桌面版（PC/Mac）便是基于 OneRing 完成的，HTML5 新增的<audio>元素与相应的 API 在其中发挥了重要的作用，相信很快就会和大家见面。OneRing 项目主页：http://code.google.com/p/onering-desktop/

豆瓣电台桌面版（**Mac**）界面截图

浏览器扩展

Douban Pulse 是一个 Chrome 浏览器的豆瓣扩展，它可以让用户随时关注友邻动态、参与互动，同时收听"豆瓣电台"。整个扩展完全基于豆瓣 API 并结合 HTML5 与 CSS3 特性完成，除了使用代码绘制界面，CSS Animation 完成简单动画、<audio>元素实现电台外，还使用了 Web Storage 与 Web Database 进行持久化本地存储。

比如利用 Local Storage 来保存登录信息与各种状态，用以再次开启扩展时迅速恢复关闭前的界面；之外还用于实时保存用户输入的文字信息，确保信息不会因网络中断、浏览器崩溃而丢失。 比起 Cookie，Local Storage 的优势显而易见：存储空间大、数据不会随请求发送至服务器，存储空间独立等等，你还可以利用浏览器原生的 JSON 支持，用它来保存结构化对象。

Douban Pulse 偏好设置页界面截图

Douban Pulse 广播列表界面截图

Douban Pulse 的歌曲信息是保存在 Web Database 中的，虽然这种使用 SQL 语句通过 Web SQL Database API 操作数据的方式在 HTML5 规范中存在一些争议(W3C 也停止了该文档的维护)，但书写 SQL 存储复杂数据的方式还是让前端开发者眼前一亮，Chrome 从 4.0 开始至今对其有着良好的支持。

Douban Pulse 电台已播放曲目列表截图

手机应用

如今 Webkit 在智能手机平台大行其道，iOS 中的 Safari Mobile、Android 的 Chrome Lite 都是很好的实践者，这也为基于 HTML5 的 web app 的开发提供了便利，利用一些诸如 PhoneGap 的框架还可以轻松地将 web app 变成 native app 上架 App Store。我也尝试了这种方式并将 Douban Pulse 成功地移植到了 iOS 上：

Douban Pulse iPhone 版（**Labs** 产品，非官方应用）界面截图

结束语

当然除此之外还有很多激动人心的 HTML5 特性没有提及，比如通信方面的 Web Sockets、Web Workers、桌面提醒 Notifications 以及倍受 LBS 应用青睐的 Geolocation 等等，大家在日常开发中也可以尝试使用，享受 HTML5 为我们带来的惊喜与便利。

虽然正式标准预计要 2022 年才正式发布，但目前 HTML5 已经被广泛实现，随着规范的不断完善、浏览器支持度的提高，其优势也愈加明显，也许会成为未来 web 应用的核心。

关于作者

石岩，豆瓣前端工程师，关注网站前端开发、移动设备应用开发，他的豆瓣主页：http://douban.com/people/mockee。

另外，感谢豆瓣网张克军先生对 InfoQ 中文站的大力支持。

特别专题

使用 HTML5 构建下一代的 Web Form

作者 蒋博

HTML5 是由 WHATWG (Web Hypertext Application Technology Working Group) 发起的，最开始的名称叫做 Web Application 1.0，而后这个标准吸纳了 Web Forms 2.0 的标准，并一同被 W3C 组织所采用，合并成为下一代的 HTML5 标准。

前言

HTML 语言作为如今编程最为广泛的语言，具有易用、快捷、多浏览平台兼容等特点，但是随着时代的进步，HTML 的标准却停滞不前，这一次还在不断开发中的 HTML5 标准的更新可以说给这门标记语言带来了新的生命力。本文将着重讨论 HTML5 中的 Web Forms 2.0, 即表单的部分。

表单是网页中常见的控件(集)。小到网站注册登录，大到一个企业的数据管理系统，都基本上有表单的身影。表单之所以如此重要，主要是因为它担负大量的用户和网页后台数据更新交互的任务。Web 开发人员，对于网页表单可以说又爱又恨，爱的是它方便的收集、组织数据的功能，恨的是它的功能很大程度上也就仅此而已。一些在最终网站用户看起来稀松平常的功能，比如说输入类型检查、表单校验、错误提示等等，开发人员无不需要花费大量精力利用 JavaScript 和 DOM 编程来满足这些天然所需的功能点，而随着 Ajax 的流行，出现的一些 JavaScript 的工具库，比如 Dojo, YUI 等都提供了方便的 JavaScript Widget 或者 API 来减轻开发人员的负担。

HTML5 的表单新特性

HTML5 Web Forms 2.0 是对目前 Web 表单的全面提升，它在保持了简便易用的特性的同时，增加了许多内置的控件或者控件属性来满足用户的需求，并且同时减少了开发人员的编程。在我看来，HTML5 主要在以下几个方面对目前的 Web 表单做了改进：

- 新的控件类型

  ✧ 还在为类型检查犯愁吗，还在为那一长串看不太明白的检验输入的正则表达式而苦恼吗，HTML5 提供的一系列新的控件将天然的具备类型检查的功能。比如说 URL 输入框，Email 输入框等。

```
<input type="url"></input>
<input type="email"></input>
```

  ✧ 当然还有非常重要的日期输入框，要知道使用 JavaScript 和 CSS 来“手工”制作一个日期输入框还是非常花功夫的，类似 Dojo，YUI 这样的类库也无不在这个 widget 上面大做文章。

```
<input type="date"></input>
```

  ✧ 作为我痛苦记忆的一部分，我经常记得我们开发人员要为一个 select 下拉别表动态的添加非常多的选项,这些选项大多数都是来自数据库，比如说国家、省市列表等等。这个事情非常繁琐。HTML5 将支持 data 属性，为 select 控件外联数据源！

```
<select data="http://domain/getmyoptions"></select>
```

- 改进的文件上传控件，你可以使用一个控件上传多个文件，自行规定上传文件的类型（accept），你甚至可以设定每个文件最大的大小（maxlength）。你看出它和一般操作系统提供的文件上传控件的区别了吗，反正我觉得基本一致了。在 HTML5 应用中，文件上传控件将变得非常强大和易用。

- 重复（repeat）的模型，HTML5 提供一套重复机制来帮助我们构建一些重复输入列表，其中包括一些诸如 add、remove、 move-up，move-down 的按钮类型，通过这一套重复的机制，开发人员可以非常方便的实现我们经常看到的编辑列表，这是一个很常见的模式，我们可以增加一个条目、删除某个条目、或者移动某个条目等等。

- 内建的表单校验系统，HTML5 为不同类型的输入控件各自提供了新的属性，来控制这些控件的输入行为，比如我们常见的必填项 required 属性，以及为数字类型控件提供的 max、min 等。 而在你提交表单的时候，一旦校验错误，浏览器将不执行提交操作，而会显示相应的检验错误信息。

```
<input type="text" required></input>
<input type="number" min=10  max=100></input>
```

- XML Submission，我们一般常见的是 form 的编码格式是 application/x-www-form-urlencoded。开发人员都很清楚这种格式，数据送到服务器端，可以方便的存取。HTML5 将提供一种新的数据格式：XML Submission，即 application/x-www-form+xml。简单的举例说，服务器端将直接接收到 XML 形式的表单

数据。

```
<submission>
    <field name="name" index="0">Peter</field>
    <field name="password" index="0">password</field>
</submission>
```

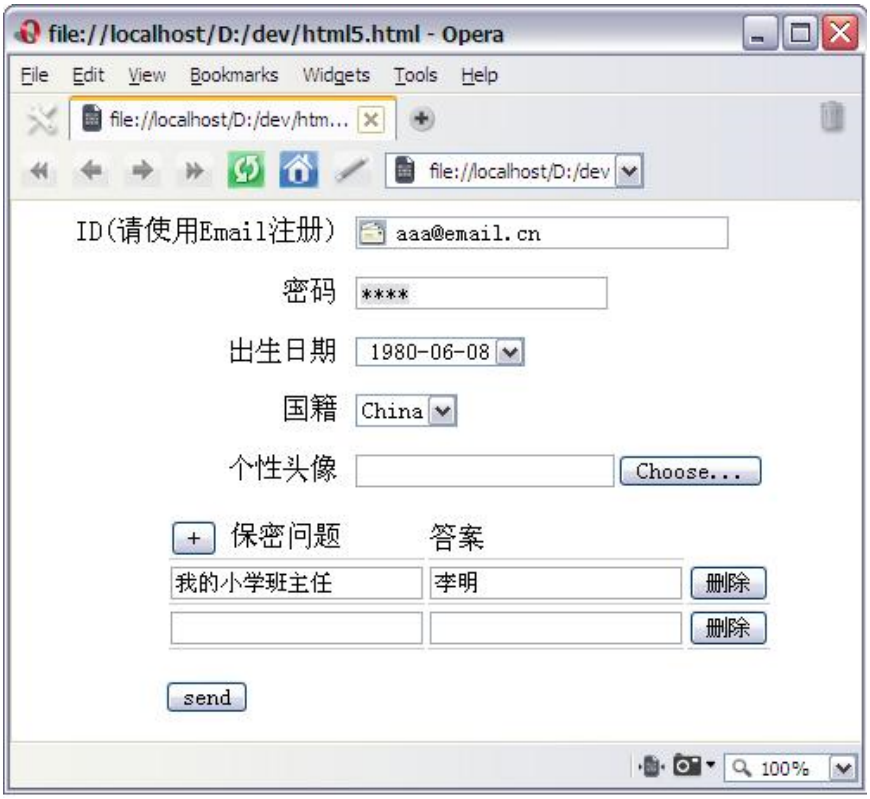
实例分析

我将利用 HTML5 新的表单系统，做一个简单的用户注册的界面，包括用户名，密码，出生日期，保密问题等内容，代码如下：

```
<! doctype html>
<html>
    <head>
        <style>
            p label {
                width: 180px;
                float: left;
                text-align: right;
                padding-right: 10px
            }
            table {
                margin-left: 80px
            }
            table td {
                border-bottom: 1px solid #CCCCCC
            }
            input.submit {
                margin-left: 80px
            }
        </style>
    </head>
    <body>
        <form action='/register'
            enctype="application/x-www-form+xml" method="post">
            <p>
                <label for='name'>ID(请使用Email注册)</label>
                <input name='name' required type='email'></input>
            </p>
            <p>
                <label for='password'>密码</label>
                <input name='password' required type='password'></input>
            </p>
            <p>
                <label for='birthday'>出生日期</label>
                <input type='date' name='birthday' />
            </p>
            <p>
                <label for='gender'>国籍</label>
                <select name='country' data='countries.xml'></select>
```

```
            </p>
            <p>
                <label for='photo'>个性头像</label>
                <input type='file' name='photo' accept='image/*' />
            </p>
            <table>
                <thead>
                    <td><button type="add" template="questionId">+</button>
                        保密问题</td>
                    <td>答案</td>
                    <td></td>
                </thead>
                <tr id="questionId" repeat="template"
                   repeat-start="1" repeat-min="1" repeat-max="3">
                    <td><input type="text" name="questions[questionId].q"></td>
                    <td><input type="text" name="questions[questionId].a"></td>
                    <td><button type="remove">删除</button></td>
                </tr>
            </table>
            <p>
                <input type='submit' value='send' class='submit' />
            </p>
        </form>
    </body>
</html>
```

由于目前 HTML5 标准仍然在开发中，不同的浏览器对 HTML5 特性的支持都相当有限。其中 Opera 在表单方面支持得比较好，本实例在 Opera9 上运行一切正常，效果图如下：



这个实例运用了一些 HTML5 的新的表单元素，比如 email 类型的输入框(ID)，日期类型的输入框(出生日期)。并且使用了重复模型来引导用户填写保密问题，而在个性头像的上传中，通过限制文件类型，方便用户选择图片进行合乎规范的内容上传。而用户选择国籍的下拉选择输入框中，采用的是外联数据源的形式，外联数据源使用 coutries.xml，内容如下：

```
<select xmlns="http://www.w3.org/1999/xhtml">
    <option>China</option>
    <option>Japan</option>
    <option>Korea</option>
</select>
```

并且 form 的 enctype 是 application/x-www-form+xml，也就是 HTML5 的 XML 提交。所以一旦 form 校验通过，form 的内容将会以 XML 的形式提交。你还会发现，在 ID 输入框如果没有值，或者输入了非 email 类型的字符串时，一旦试图提交表单，就会有提示错误的信息出现，而这都是浏览器内置的。

结语

HTML5 对表单控件的更新，无疑是很振奋人心的。本文描述了一部分表单的新特性，还有一部分新特性同样很令人期待。相信随着标准的深入开发以及浏览器对 HTML5 支持程度的进一步提升，设计一个简单易用的表单的工作，将变得非常轻松。

参考资料

1. W3C HTML5 规范
2. HTML5 与 HTML4 的差异
3. Opera dev: Improve your forms with HTML5
4. Comparison of layout engines（HTML5）

关于作者

蒋博，主要从事 Web 前端技术的开发工作，在 Web 开发与性能分析以及敏捷实践等领域有较丰富的经验。对 HTML5 的发展以及各种 JavaScript 类 库有比较浓厚的兴趣，经常关注社交型的网站发展情况，平常喜欢听音乐，看一些历史类书籍。（本文仅代表个人观点，与公司立场无关。）

原文链接：http://www.infoq.com/cn/articles/html5-web-form

加速你的 Hibernate 引擎（下）

作者 Yongjun Jiao and Stewart Clark 译者 丁雪丰

（接上文）

4.6 HQL 调优

4.6.1 索引调优

HQL 看起来和 SQL 很相似。从 HQL 的 WHERE 子句中通常可以猜到相应的 SQL WHERE 子句。WHERE 子句中的字段决定了数据库将选择的索引。

大多数 Hibernate 开发者所常犯的一个错误是无论何时，当需要新 WHERE 子句的时候都会创建一个新的索引。因为索引会带来额外的数据更新开销，所以应该争取创建少量索引来覆盖尽可能多的查询。

4.1 节让你使用一个集合来处理所有可能的数据搜索条件。如果这不太实际，那么你可以使用后端剖析工具来创建一个针对应用程序涉及的所有 SQL 的集合。基于那些搜索条件的分类，你最终会得到一个小的索引集。与此同时，还可以尝试向 WHERE 子句中添加额外的谓语来匹配其他 WHERE 子句。

> 范例 7
>
> 有两个 UI 搜索器和一个后端守护进程搜索器来搜索名为 iso_deals 的表。第一个 UI 搜索器在 unexpectedFlag、dealStatus、tradeDate 和 isold 属性上有谓语。
>
> 第二个 UI 搜索器基于用户键入的过滤器，其中包括的内容除 tradeDate 和 isold 以外还有其他属性。开始时所有这些过滤器属性都是可选的。
>
> 后端搜索器基于 isold、participantCode 和 transactionType 属性。
>
> 经过进一步业务分析，发现第二个 UI 搜索器实际是基于一些隐式的 unexpectedFlag 和 dealStatus 值来选择数据的。我们还让 tradeDate 成为过滤器的必要属性（为了使

用数据库索引，每个搜索过滤器都应该有必要属性）。

鉴于这一点，我们依次使用 unexpectedFlag、dealStatus、tradeDate 和 isold 构造了一个复合索引。两个 UI 搜索器都能共用它。（顺序很重要，如果你的谓语以不同的顺序指定这些属性或在它们前罗列了其他属性，数据库就不会选择该复合索引。）

后端搜索器和 UI 搜索器区别太大，因此我们不得不为它构造另一个复合索引，依次使用 isold、participantCode 和 transactionType。

4.6.2 绑定参数 **vs.**字符串拼接

既可以使用绑定参数构造 HQL 的 WHERE 子句，也可以使用字符串拼接的方法，该决定对性能会有一定影响。使用绑定参数的原因是让数据库一次解析 SQL，对后续的重复请求复用生成好的执行计划，这样做节省了 CPU 时间和内存。然而，为达到最优的数据访问效率，不同的绑定值可能需要不同的 SQL 执行计划。

例如，一小段数据范围可能只返回数据总量的 5%，而一大段数据范围可能返回数据总量的 90%。前者使用索引更好，而后者则最好使用全表扫描。

建议 OLTP 使用绑定参数，数据仓库使用字符串拼接，因为 OLTP 通常在一个事务中重复插入和更新数据，只取少量数据；数据仓库通常只有少量 SQL 查询，有一个确定的执行计划比节省 CPU 时间和内存更为重要。

要是你知道你的 OLTP 搜索对不同绑定值应该使用相同执行计划又该怎么办呢？

Oracle 9i 及以后版本在第一次调用绑定参数并生成执行计划时能探出参数值。后续调用不会再探测，而是重用之前的执行计划。

4.6.3 聚合及排序

你可以在数据库中进行聚合和“order by”，也可以在应用程序的服务层中事先加载所有数据然后做聚合和“order by”操作。推荐使用前者，因为数据库在这方面通常会比你的应用程序做得好。此外，这样做还能节省网络带宽，这也是一种拥有跨数据库移植性的做法。

当你的应用程序对数据聚合和排序有 HQL 不支持的特定业务规则时除外。

4.6.4 覆盖抓取策略

详见 **4.7.1** 节。

4.6.5 本地查询

本地查询调优其实并不直接与 HQL 有关。但 HQL 的确可以让你直接向底层数据库传递本地查询。我们并不建议这么做，因为本地查询在数据库间不可移植。

4.7 抓取策略调优

抓取策略决定了在应用程序需要访问关联对象时，Hibernate 以何种方式以及何时获取关联对象。HRD 中的第 20 章"改善性能"对该主题作了很好的阐述，我们在此将关注它的使用方法。

4.7.1 覆盖抓取策略

不同的用户可能会有不同的数据抓取要求。Hibernate 允许在两个地方定义数据抓取策略，一处是在映射元数据中，另一处是在 HQL 或 Criteria 中覆盖它。

常见的做法是基于主要的抓取用例在映射元数据中定义默认抓取策略，针对少数用例在 HQL 和 Criteria 中覆盖抓取策略。

假设 pojoA 和 pojoB 是父子关系实例。如果根据业务规则，只是偶尔需要从实体两端加载数据，那你可以声明一个延迟加载集合或代理抓取（proxy fetching）。当你需要从实体两端获取数据时，可以用立即抓取（eager fetching）覆盖默认策略，例如使用 HQL 或 Criteria 配置连接抓取（join fetching）。

另一方面，如果业务规则在大多数时候需要从实体两端加载数据，那么你可以声明立即抓取并在 Criteria 中设置延迟加载集合或代理抓取来覆盖它（HQL 目前还不支持这样的覆盖）。

4.7.2 N+1 模式或是反模式？

select 抓取会导致 N+1 问题。如果你知道自己总是需要从关联中加载数据，那么就该始终使用连接抓取。在下面两个场景中，你可能会把 N+1 视为一种模式而非反模式。

第一种场景，你不知道用户是否会访问关联对象。如果他/她没有访问，那么你赢了；否则你仍然需要额外的 N 次 select SQL 语句。这是一种令人左右为难的局面。

第二种场景，pojoA 和很多其他 POJO 有 one-to-many 关联，例如 pojoB 和 pojoC。使用立即的内连接或外连接抓取会在结果集中将 pojoA 重复很多次。当 pojoA 中有很多非空属性时，你不得不将大量数据加载到持久层中。这种加载需要很多时间，既有网络带宽的原因，如果 Hibernate 的会话是有状态的，其中也会有会话缓存的原因（内存消耗和 GC 暂停）。

如果你有一个很长的 one-to-many 关联链，例如从 pojoA 到 pojoB 到 pojoC 以此类推，情况

也是类似的。

你也许会去使用 HQL 中的 DISTINCT 关键字或 Cirteria 中的 distinct 功能或是 Java 的 Set 接口来消除重复数据。但所有这些都是在 Hibernate（在持久层）中实现的，而非数据库中。

如果基于你的网络和内存配置的测试表明 N+1 性能更好，那么你可以使用批量抓取、subselect 抓取或二级缓存来做进一步调优。

> 范例 **8**
>
> 以下是一个使用批量抓取的 HBM 文件片段：
>
> ```
> <class name="pojoA" table="pojoA">
>  …
>  <set name="pojoBs" fetch="select" batch-size="10">
>
>  <key column="pojoa_id"/>
>  …
>  </set>
> </class>
> ```
>
> 以下是多端 pojoB 生成的 SQL：
>
> ```
> select … from pojoB where pojoa_id in(?,?,?,?,?, ?,?,?,?,?);
> ```
>
> 问号数量与 batch-size 值相等。因此 N 次额外的关于 pojoB 的 select SQL 语句被减少到了 N/10 次。
>
> 如果将 fetch="select"替换成 fetch="subselect"，pojoB 生成的 SQL 语句就是这样的：
>
> ```
> select … from pojoB where pojoa_id in(select id from pojoA where …);
> ```
>
> 尽管 N 次额外的 select 减少到 1 次，但这只在重复运行 pojoA 的查询开销很低时才有好处。
>
> 如果 pojoA 中的 pojoB 集合很稳定，或 pojoB 有 pojoA 的 many-to-one 关联，而且 pojoA 是只读引用数据，那么你可以使用二级缓存来缓存 pojoA 以消除 N+1 问题（**4.8.1** 节中有一个例子）。

4.7.3 延迟属性抓取

除非有一张拥有很多你不需要的字段的遗留表，否则不应该使用这种抓取策略，因为它的延迟属性分组会带来额外的 SQL。

在业务分析和设计过程中，你应该将不同数据获取或修改分组放到不同的领域对象实体中，而不是使用这种抓取策略。

如果不能重新设计遗留表，可以使用 HQL 或 Criteria 提供的投影功能来获取数据。

4.8 二级缓存调优

HRD 第 20.2 节 “二级缓存”中的描述对大多数开发者来说过于简单，无法做出选择。3.3 版及以后版本不再推荐使用基于“CacheProvider”的缓存，而用基于“RegionFactory”的缓存，这也让人更糊涂了。但是就算是最新的 3.5 参考文档也没有提及如何使用新缓存方法。

出于下述考虑，我们将继续关注于老方法：

- 所有流行的 Hibernate 二级缓存提供商中只有 JBoss Cache 2、Infinispan 4 和 Ehcache 2 支持新方法。OSCache、SwarmCache、Coherence 和 Gigaspaces XAP-Data Grid 只支持老方法。
- 两种方法共用相同的<cache>配置。例如，它们仍旧使用相同的 usage 属性值“transactional|read-write|nonstrict-read-write|read-only”。
- 多个 cache-region 适配器仍然内置老方法的支持，理解它能帮助你快速理解新方法。

4.8.1 基于 CacheProvider 的缓存机制

理解该机制是做出合理选择的关键。关键的类/接口是 CacheConcurrencyStrategy 和它针对 4 中不同缓存使用的实现类，还有 EntityUpdate/Delete/InsertAction。

针对并发缓存访问，有三种实现模式：

- 针对“**read-only**”的只读模式。

  无论是锁还是事务都没影响，因为缓存自数据从数据库加载后就不会改变。

- 针对“**read-write**”和“**nonstrict-read-write**”的非事务感知（**non-transaction-aware**）读写模式。

  缓存的更新发生在数据库事务完成后。缓存需要支持锁。

- 针对“**transactional**”的事务感知读写。

  对缓存和数据库的更新被包装在同一个 JTA 事务中，这样缓存与数据库总是保持同步的。数据库和缓存都必须支持 JTA。尽管缓存事务内部依赖于缓存锁，但 Hibernate 不会显式调用任何的缓存锁函数。

以数据库更新为例。EntityUpdateAction 对于事务感知读写、“read-write”的非事务感知读写，

还有“nonstrict-read-write”的非事务感知读写相应有如下调用序列：

- 在一个 **JTA** 事务中更新数据库；在同一个事务中更新缓存。
- 软锁缓存；在一个事务中更新数据库；在上一个事务成功完成后更新缓存；否则释放软锁。

  软锁只是一种特定的缓存值失效表述方式，在它获得新数据库值前阻止其他事务读写缓存。那些事务会转而直接读取数据库。

  缓存必须支持锁；事务支持则不是必须的。如果缓存是一个集群，“更新缓存”的调用会将新值推送给所有副本，这通常被称为“推（push）”更新策略。
- 在一个事务中更新数据库；在上一个事务完成前就清除缓存；为了安全起见，无论事务成功与否，在事务完成后再次清除缓存。

  既不需要支持缓存锁，也不需要支持事务。如果是缓存集群，“清除缓存”调用会让所有副本都失效，这通常被称为“拉（pull）”更新策略。

对于实体的删除或插入动作，或者集合变更，调用序列都是相似的。

实际上，最后两个异步调用序列仍能保证数据库和缓存的一致性（基本就是“read committed”的隔离了级别），这要归功于第二个序列中的软锁和“更新数据库”后的“更新缓存”，还有最后一个调用序列中的悲观“清除缓存”。

基于上述分析，我们的建议是：

- 如果数据是只读的，例如引用数据，那么总是使用“read-only”策略，因为它是最简单、最高效的策略，也是集群安全的策略。
- 除非你真的想将缓存更新和数据库更新放在一个 JTA 事务里，否则不要使用“transactional”策略，因为 JTA 需要漫长的两阶段提交处理，这导致它基本是性能最差的策略。

  依笔者看来，二级缓存并非一级数据源，因此使用 JTA 也未必合理。实际上最后两个调用序列在大多数场景下是个不错的替代方案，这要归功于它们的数据一致性保障。
- 如果你的数据读很多或者很少有并发缓存访问和更新，那么可以使用“nonstrict-read-write”策略。感谢它的轻量级“拉”更新策略，它通常是性能第二好的策略。

- 如果你的数据是又读又写的，那么使用“read-write”策略。这通常是性能倒数第二的策略，因为它要求有缓存锁，缓存集群中使用重量级的“推”更新策略。

范例 9

以下是一个 ISO 收费类型的 HBM 文件片段：

```
<class name="IsoChargeType">
      <property name="isoId" column="ISO_ID" not-null="true"/>
      <many-to-one name="estimateMethod" fetch="join" lazy="false"/>
      <many-to-one  name="allocationMethod" fetch="join"
         lazy="false"/>
      <many-to-one name="chargeTypeCategory" fetch="join"
         lazy="false"/>
</class>
```

一些用户只需要 ISO 收费类型本身；一些用户既需要 ISO 收费类型，还需要它的三个关联对象。简单起见，开发者会立即加载所有三个关联对象。如果项目中没人负责 Hibernate 调优，这是很常见的。

4.7.1 节中讲过了最好的方法。因为所有的关联对象都是只读引用数据，另一种方法是使用延迟抓取，打开这些对象的二级缓存以避免 N+1 问题。实际上前一种方法也能从引用数据缓存中获益。

因为大多数项目都有很多被其他数据引用的只读引用数据，上述两种方法都能改善全局系统性能。

4.8.2 RegionFactory

下表是新老两种方法中对应的主要类/接口：

新方法	老方法
RegionFactory	CacheProvider
Region	Cache
EntityRegionAccessStrategy	CacheConcurrencyStrategy
CollectionRegionAccessStrategy	CacheConcurrencyStrategy

第一个改进是 RegionFactory 构建了特定的 Region，例如 EntityRegion 和 TransactionRegion，而不是使用一个通用的访问 Region。第二个改进是对于特定缓存的“usage”属性值，Region 要求构建自己的访问策略，而不是所有 Region 都一直使用 CacheConcurrencyStrategy 的 4 种

实现。

要使用新方法，应该设置 factory_class 而非 provider_class 配置属性。以 Ehcache 2.0 为例：

```
<property name="hibernate.cache.region.factory_class">
       net.sf.ehcache.hibernate.EhCacheRegionFactory
</property>
```

其他相关的 Hibernate 缓存配置都和老方法一样。

新方法也能向后兼容遗留方法。如果还是只配了 CacheProvider，新方法中将使用下列自说明（self-explanatory）适配器和桥隐式地调用老的接口/类：

RegionFactoryCacheProviderBridge、EntityRegionAdapter、CollectionRegionAdapter、QueryResultsRegionAdapter、EntityAccessStrategyAdapter 和 CollectionAccessStrategyAdapter

4.8.3 查询缓存

二级缓存也能缓存查询结果。如果查询开销很大而且要重复运行，这也会很有帮助。

4.9 批量处理调优

大多数 Hibernate 的功能都很适合那些每个事务都通常只处理少量数据的 OLTP 系统。但是，如果你有一个数据仓库或者事务需要处理大量数据，那么就另当别论了。

4.9.1 使用有状态会话的非 DML 风格批处理

如果你已经在使用常规会话了，那这是最自然的方法。你需要做三件事：

- 配置下列 3 个属性以开启批处理特性：

  ```
  hibernate.jdbc.batch_size 30
  hibernate.jdbc.batch_versioned_data true
  hibernate.cache.use_second_level_cache false
  ```

  batch_size 设置为正值会开启 JDBC2 的批量更新，Hibernate 的建议值是 5 到 30。基于我们的测试，极低值和极高值性能都很差。只要取值在合理范围内，区别就只有几秒而已。如果网络够快，这个结果是一定的。

  第二个配置设为 true，这要求 JDBC 驱动在 executeBatch()方法中返回正确的行数。对于 Oracle 用户而言，批量更新时不能将其设为 true。请阅读 Oracle 的《JDBC Developer's Guide and Reference》中的"标准批处理的 Oracle 实现中的更新计数"（Update Counts in the Oracle Implementation of Standard Batching）以获得更多详细信息。因为它对批量插入来说还是安全的，所以你可以为批量插入创建单独的专用数据源。最后一个配置项是可选

的，因为你可以在会话中显式关闭二级缓存。

- 像如下范例中那样定期刷新（flush）并清除一级会话缓存：

```
   Session session = sessionFactory.openSession();
   Transaction tx = session.beginTransaction();
   for ( int i=0; i<100000; i++ ) {
      Customer customer = new Customer(.....);
      //if your hibernate.cache.use_second_level_cache is true, call the
following:
      session.setCacheMode(CacheMode.IGNORE);
      session.save(customer);
      if (i % 50 == 0) { //50, same as the JDBC batch size
         //flush a batch of inserts and release memory:
         session.flush();
         session.clear();
       }
   }
   tx.commit();
   session.close();
```

批处理通常不需要数据缓存，否则你会将内存耗尽并大量增加 GC 开销。如果内存有限，那这种情况会很明显。

- 总是将批量插入嵌套在事务中。

每次事务修改的对象数量越少就意味着会有更多数据库提交，正如 **4.5** 节所述每次提交都会带来磁盘相关的开销。

另一方面，每次事务修改的对象数量越多就意味着锁定变更时间越长，同时数据库需要更大的 redo log。

4.9.2 使用无状态会话的非 **DML** 风格批处理

无状态会话执行起来比上一种方法更好，因为它只是 JDBC 的简单包装，而且可以绕开很多常规会话要求的操作。例如，它不需要会话缓存，也不和任何二级缓存或查询缓存有交互。

然而它的用法并不简单。尤其是它的操作并不会级联到所关联的实例上；你必须自己来处理它们。

4.9.3 DML 风格

使用 DML 风格的插入、更新或删除，你直接在数据库中操作数据，这和前两种方法在 Hibernate 中操作数据的情况有所不同。

因为一个 DML 风格的更新或删除相当于前两种方法中的多个单独的更新或删除，所以如果更新或删除中的 WHERE 子句暗示了恰当的数据库索引，那么使用 DML 风格的操作能节省网

络开销，执行得更好。

强烈建议结合使用 DML 风格操作和无状态会话。如果使用有状态会话，不要忘记在执行 DML 前清除缓存，否则 Hibernate 将会更新或清除相关缓存（见下面的范例 **10**）。

4.9.4 批量加载

如果你的 HQL 或 Criteria 会返回很多数据，那么要注意两件事：

- 用下列配置开启批量抓取特性：

```
hibernate.jdbc.fetch_size 10
```

  fetch_size 设置为正值将开启 JDBC 批量抓取特性。相对快速网络，在慢速网络中这一点更为重要。Oracle 建议的经验值是 10。你应该基于自己的环境进行测试。

- 在使用上述任一方法时都要关闭缓存，因为批量加载一般是一次性任务。受限于内存容量，向缓存中加载大量数据通常也意味着它们很快会被清除出去，这会增加 GC 开销。

> 范例 **10**
>
> 我们有一个后台任务，分段加载大量的 IsoDeal 数据用于后续处理。我们还会在分段数据交给下游系统处理前将其更新为处理中状态。最大的一段有 50 万行数据。以下是原始代码中截取出来的一段：
>
> ```
> Query query = session.createQuery(
>  "FROM IsoDeal d WHERE chunk-clause");
> query.setLockMode("d", LockMode.UPGRADE); //for Inprocess status update
> List<IsoDeal> isoDeals = query.list();
> for (IsoDeal isoDeal : isoDeals) { //update status to Inprocess
>  isoDeal.setStatus("Inprocess");
> }
> return isoDeals;
> ```
>
> 包含上述代码的方法加上了 Spring 2.5 声明式事务的注解。加载并更新 50 万行数据大约花了 10 分钟。我们识别出了以下这些问题：
>
> - 由于会话缓存和二级缓存的原因，系统会频繁地内存溢出。
> - 就算没有内存溢出，当内存消耗很高时 GC 的开销也会很大。
> - 我们还未设置 fetch_size。
> - 就算我们设置了 batch_size，for 循环也创建了太多 update SQL 语句。
>
> 不幸的是 Spring 2.5 不支持 Hibernate 无状态会话，所以我们只能关闭二级缓存；设

置 fetch_size；用 DML 风格的更新来代替 for 循环，以此改善性能。

但是，执行时间还是要 6 分钟。将 Hibernate 的日志级别调成 trace 后，我们发现是更新会话缓存造成了延时。通过在 DML 更新前清除会话缓存，我们将时间缩短到了 4 分钟，全部都是将数据加载到会话缓存中花费的时间。

4.10 SQL 生成调优

本节将向你展示如何减少 SQL 生成的数量。

4.10.1 N+1 抓取问题

“select 抓取”策略会导致 N+1 问题。如果“连接抓取”策略适合你的话，你应该始终使用该策略避免 N+1 问题。

但是，如果“连接抓取”策略执行效果不理想，就像 **4.7.2** 节中那样，你可以使用“subselect 抓取”、“批量抓取”或“延迟集合抓取”来减少所需的额外 SQL 语句数。

4.10.2 Insert+Update 问题

> 范例 **11**
>
> 我们的 ElectricityDeal 与 DealCharge 有单向 one-to-many 关联，如下列 HBM 文件片段所示：

```
<class name="ElectricityDeal"
    select-before-update="true" dynamic-update="true"
    dynamic-insert="true">
    <id name="key" column="ID">
        <generator class="sequence">
            <param name="sequence">SEQ_ELECTRICITY_DEALS</param>
        </generator>
    </id>
    …
    <set name="dealCharges" cascade="all-delete-orphan">
        <key column="DEAL_KEY" not-null="false" update="true"
                on-delete="noaction"/>
        <one-to-many class="DealCharge"/>
    </set>
</class>
```

在“key”元素中，“not-null”和“update”对应的默认值是 false 和 true，上述代码为了明确这些取值，将它们写了出来。

如果你想创建一个 ElectricityDeal 和十个 DealCharge，会生成如下 SQL 语句：

- 1 句 ElectricityDeal 的插入语句；
- 10 句 DealCharge 的插入语句，其中不包括外键“DEAL_KEY”；
- 10 句 DealCharge 字段“DEAL_KEY”的更新语句。

为了消除那额外的 10 句更新语句，可以在那 10 句 DealCharge 插入语句中包含“DEAL_KEY”，你需要将“not-null”和“update”分别修改为 true 和 false。

另一种做法是使用双向或 many-to-one 关联，让 DealCharge 来管理关联。

4.10.3 更新前执行 select

在范例 **11** 中，我们为 ElectricityDeal 加上了 select-before-update，这会对瞬时（transient）对象或分离（detached）对象产生额外的 select 语句，但却能避免不必要的数据库更新。

你应该做出一些权衡，如果对象没多少属性，不需要防止不必要的数据库更新，那么就不要使用该特性，因为你那些有限的数据既没有太多网络传输开销，也不会带来太多数据库更新开销。

如果对象的属性较多，例如是一张大的遗留表，那你应该开启该特性，和“dynamic-update”结合使用以避免太多数据库更新开销。

4.10.4 级联删除

在范例 11 中，如果你想删除 1 个 ElectricityDeal 和它的 100 个 DealCharge，Hibernate 会对 DealCharge 做 100 次删除。

如果将“on-delete”修改为“cascade”，Hibernate 不会执行 DealCharge 的删除动作；而是让数据库根据 ON CASCADE DELETE 约束自动删除那 100 个 DealCharge。不过，需要让 DBA 开启 ON CASCADE DELETE 约束，大多数 DBA 不愿意这么做，因为他们想避免父对象的意外删除级联到它的依赖对象上。此外，还要注意，该特性会绕过 Hibernate 对版本数据（versioned data）的常用乐观锁策略。

4.10.5 增强的序列标识符生成器

范例 **11** 中使用 Oracle 的序列作为标识符生成器。假设我们保存 100 个 ElectricityDeal，Hibernate 会将下面的 SQL 语句执行 100 次来获取下一个可用的标识符：

```
select SEQ_ELECTRICITY_DEALS.NEXTVAL from dual;
```

如果网络不是很快，那这无疑会降低效率。3.2.3 及后续版本中增加了一个增强的生成器“SequenceStyleGenerator”，它带了两个优化器：hilo 和 pooled。尽管 HRD 的第 5 章“基础 O/R

映射” 讲到了这两个优化器，不过内容有限。两个优化器都使用了 HiLo 算法，该算法生成的标识符等于 Hi 值加上 Lo 值，其中 Hi 值代表组号，Lo 值顺序且重复地从 1 迭代到最大组大小，组号在 Lo 值“转回到”1 时加 1。

假设组大小是 5（可以用 max_lo 或 increment_size 参数来表示），下面是个例子：

Group Number	0					1					2				
Hi value	0					5					10				
Lo value	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Identifier value	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

- hilo 优化器

  组号取自数据库序列的下一个可用值，Hi 值由 Hibernate 定义，是组号乘以 increment_size 参数值。

- pooled 优化器

  Hi 值直接取自数据库序列的下一个可用值。数据库序列的增量应该设置为 increment_size 参数值。

直到内存组中的值耗尽后，两个优化器才会去访问数据库，上面的例子每 5 个标识值符访问一次数据库。使用 hilo 优化器时，你的序列不能再被其他应用程序使用，除非它们使用与 Hibernate 相同的逻辑。使用 pooled 优化器，在其他应用程序使用同一序列时则相当安全。

两个优化器都有一个问题，如果 Hibernate 崩溃，当前组内的一些标识符值就会丢失，然而大多数应用程序都不要求拥有连续的标识符值（如果你的数据库，比方说 Oracle，缓存了序列值，当它崩溃时你也会丢失标识符值）。

如果在范例 11 中使用 pooled 优化器，新的 id 配置如下：

```
<id name="key" column="ID">
  <generator class="org.hibernate.id.enhance.SequenceStyleGenerator">
    <param name="sequence_name">SEQ_ELECTRICITY_DEALS</param>

    <param name="initial_value">0</param>
    <param name="increment_size">100</param>

    <param name="optimizer ">pooled</param>
  </generator>
</id>
```

5 总结

本文涵盖了大多数你在 Hibernate 应用程序调优时会觉得很有用的调优技巧，其中的大多数

时间都在讨论那些行之有效却缺乏文档的调优主题，例如继承映射、二级缓存和增强的序列标识符生成器。

它还提到了一些 Hibernate 调优所必需的数据库知识。一些范例中包含了你可能遇到的问题的实际解决方案。

除此之外，值得一提的是 Hibernate 也可以和 In-Memory Data Grid（IMDG）一起使用，例如 Oracle 的 Coherance 或 GigaSpaces IMDG，这能让你的应用程序达到毫秒级别。

6 资源

[1] Latest Hibernate Reference Documentation on jboss.com

[2] Oracle 9i Performance Tuning Guide and Reference

[3] Performance Engineering on Wikipedia

[4] Program Optimization on Wikipedia

[5] Pareto Principle (the 80/20 rule) on Wikipedia

[6] Premature Optimization on acm.org

[7] Java Performance Tuning by Jack Shirazi

[8] The Law of Leaky Abstractions by Joel Spolsky

[9] Hibernate's StatisticsService Mbean configuration with Spring

[10] JProbe by Quest Software

[11] Java VisualVM

[12] Column-oriented DBMS on Wikipedia

[13] Apache DBCP BasicDataSource

[14] JDBC Connection Pool by Oracle

[15] Connection Failover by Oracle

[16] Last Resource Commit Optimization (LRCO)

[17] GigaSpaces for Hibernate ORM Users

关于作者

Yongjun Jiao 是 SunGard Consulting Services 的技术主管。过去 10 年中他一直是专业软件开发者，他的专长包括 Java SE、Java EE、Oracle 和应用程序调优。他最近的关注点是高性能计算，包括内存数据网格、并行计算和网格计算。

Stewart Clark 是 SunGard Consulting Services 的负责人。过去 15 年中他一直是专业软件开发者和项目经理，他的专长包括 Java 核心编程、Oracle 和能源交易。

原文链接：http://www.infoq.com/cn/articles/hibernate_tuning-ii

相关内容：

- 加速你的Hibernate引擎（上）
- 用Hibernate实现领域对象的自定义字段
- 使用ETags减少Web应用带宽和负载
- 演进架构中的领域驱动设计
- Alexandru Popescu谈InfoQ.com网站架构

推荐文章 | Articles

Tellurium 自动化测试框架介绍

作者 Hari Poolla 译者 吴宇

简介

Tellurium 自动化测试框架是由方剑在 2007 年 6 月创建的用于测试 Web 应用的框架，并在 2008 年 6 月移步到 Google Code 成为一个开源项目。它会定期发布版本，当前版本号是 0.7.0。项目的核心已经历时两年，并繁衍出了多个子项目，包括：UDL、Core、Engine、Widget 扩展、 Maven Archetype、Trump、Tellurium IDE、TelluriumWorks 以及参考项目。

这个框架是从 Selenium 框架发展而来，但又具有不同的测试理念。大多数 Web 测试框架，比如 Selenium，主要致力于单独的 UI 元素。而 Tellurium 恰好相反，它把多个 UI 元素看作一个 Widget 整体，并将其称作 UI module。

拿 Google 搜索的 UI 做个例子，这个界面用 Tellurium 表示成如下这样：

```
ui.Container(uid: "GoogleSearchModule", clocator: [tag: "td"]){
   InputBox(uid: "Input", clocator: [title: "Google Search"])
   SubmitButton(uid: "Search", clocator: [name: "btnG", value: "Google
Search"])
   SubmitButton(uid: "ImFeelingLucky", clocator: [value: "I'm Feeling
 Lucky"])
}
```

正如你在例子中看到的，UI module 是嵌套的 UI 元素、tag 以及 attribute 的集合。Tellurium 在采用了 UI module 之后，更具表述性，对变化的响应也更加智能化。同时它也可以很容易地表示动态 Web 内容，并易于维护。

这个框架由以下组件组成：

1. **Trump**——一个 Firefox 插件，全称是 Tellurium UI Module Plugin，它可以在用户选择待测 Web 页面上的 UI 元素后，自动生成 UI module。
2. **Tellurium IDE**——另外一个 Firefox 插件，可以记录用户操作，并生成 Tellurium 测试脚本，包括 UI module 的定义、用户的动作以及断言。测试脚本是用 Groovy 写成的。

3. TelluriumWorks——一个单独的 Java Swing 应用，用于编辑和运行 Tellurium 测试脚本。另外还在开发一个针对 IntelliJ IDEA 的 IDE 插件。
4. JavaScript Widget Extensions——针对流行 JavaScript 框架的扩展，比如 Dojo 和 jQuery UI。这样用户就可以在项目中引用发布的 Tellurium jar 文件，并在 UI module 定义中，把 UI widget 当作一个普通的 Tellurium 对象。

特性

主要特性包括：

1. UI module 清楚地表示了待测的 UI。在 Tellurium 的测试代码中，并没有直接使用定位器（locator）。而是使用 uid 对象来引用 UI 元素，从而更具表现力。比如：

```
type "GoogleSearchModule.Input", "Tellurium test"
click "GoogleSearchModule.Search"
```

2. 使用 UI 属性而不是固定的定位器来描述 UI。实际的定位器将会在运行时生成。如果属性改变了，新的运行时定位器也会由框架自动重新生成。Tellurium 从而可以按需自适应 UI 的变化。

   Tellurium 0.7.0 中的 Santa 算法通过一次定位整个的 UI module，从而进一步提高测试的智能性。此外还使用了 UI module 部分匹配机制，在一定程度上适应属性的变化。

3. 采用了 Tellurium UI 模板和 Tellurium UID 描述语言（UDL）来表示动态的 web 内容。
4. 框架实行 UI module 和测试代码的分离，从而有利于重构。

   比如，UI 和对应的测试方法定义在分离的 Groovy 类中。这样，测试代码就和 UI module 解耦了。

另外该框架还：

- 使用抽象的 UI 对象来封装 Web UI 元素
- 支持 Widget 以实现可重用性
- 为 UI 定义、动作和测试提供一门 DSL
- 支持组定位，从而一次定位一组 UI 组件
- 包括 CSS 选择器支持，以改善在 IE 中的测试速度

- 提供定位器缓存和命令集来提高测试速度
- 支持数据驱动测试

对比 Selenium 和 Tellurium

Selenium Web 测试框架是最流行的开源自动化 web 测试框架之一。它是一款独创性的框架，提供了很多独一无二的特性和优势，比如：基于浏览器的测试、Selenium Grid 以及使用 Selenium IDE 来“录制和回放”用户的动作。

然后，Selenium 有点问题。拿下面这段测试代码举个例子：

```
setUp("http://www.google.com/", "*chrome");
selenium.open("/");
selenium.type("q", "Selenium test");
selenium.click("//input[@value='Google Search' and @type='button']");
```

如果有人不是很熟悉 Google 的搜索页面，他能根据这段代码，说出页面的 UI 是什么样子的吗？定位器 q 在这里是什么意思呢？

万一因为 Web 的变化，XPath //input[@value='Google Search' and @type='button']变成无效的了，怎么办？更有可能发生的是，这段代码需要逐行检查才能找出那几行需要更新的代码。

万一这段测试代码里面有几十上百个定位器怎么办？使用 Selenium IDE 生成测试代码，这可能在一开始比较容易，但归纳和重构起来就很困难了。

重构会是一个比从头生成新测试代码更乏味的过程。原因在于硬编码的定位器和使用的测试代码耦合太紧密了。因为测试代码没有结构化，维护代码就变得异常困难。

作为一个低层次 Web 测试驱动框架，Selenium 是一个很好的框架。然而，它需要投入更多的努力才能创建出智能的测试代码。

Tellurium 就是为解决 Selenium 中的大多数问题而诞生的。

Tellurium 同时被设计用来解决 Selenium 的其他弱点。比如，IE 性能一直是 Selenium 突出的问题。Tellurium 的解决方案是，使用 CSS 选择器来作为缺省的定位器。定位器由 UI module 自动生成，并改善了测试速度。

另外，在采用了 Tellurium UI module 缓存以及基于新的 Tellurium 引擎的命令集之后，测试速度得到进一步的提升。Tellurium 还支持对 Ajax 应用的测试：Tellurium 的 List 和 Table UI 对象被用来在运行时表示动态 Web 内容。而 option 对象则被用来在运行时表示同一 Web 元素的两个不同 UI。

与 Selenium 一样，Tellurium 可以用来测试任何基于 HTML DOM 结构的 Web 应用。

测试方法

Tellurium 采用一种新的方式，通过 UI module 的概念来进行自动化测试。使用对象封装 Web UI 的元素，因此不再需要手动生成和重构 UI 的定位器。UI module 是个简单的复合 UI 对象，由嵌套的基本 UI 对象组成。

这个框架可以在两种模式下运行。第一种模式是作为 Selenium 框架的 wrapper 来工作。也就是说，Tellurium core 基于 UI module 中的 UI 对象属性，生成运行时定位器。生成的运行时定位器然后通过 Tellurium 扩展传递给 Selenium core 来调用。

Tellurium 还在开发它自己的驱动引擎，即 Tellurium Engine，以更好更有效地支持 UI module。

1. 首先，Tellurium Core 把 UI module 转换成 JSON 的表示形式。
2. 然后在使用 UI module 时，JSON 表示的数据被第一次传递给 Tellurium Engine。
3. 接着 Tellurium Engine 使用 Santa 算法，定位整个 UI module，并将其存在缓存中。
4. 在接下来的调用中，会直接使用缓存的 UI module，而不需要重新定位了。
5. 此外，Tellurium Core 把多条命令合并成一条批处理命令，叫做宏命令，然后在一次调用中把这条批处理发送给 Tellurium Engine。这样可以减少请求/响应带来的延迟。

下面这个例子，使用了该项目网站上的问题搜索 UI，来表述框架背后的思想。

我们从为问题搜索的 UI 定义 UI module 开始吧：

```
ui.Form(uid: "issueSearch", clocator: [action: "list", method: "GET"]) {
    Selector(uid: "issueType", clocator: [name: "can", id: "can", direct:
    "true"])
    TextBox(uid: "searchLabel", clocator: [tag: "span", text: "for"])
    InputBox(uid: "searchBox", clocator: [type: "text", name: "q", id: "q"])
    SubmitButton(uid: "searchButton", clocator: [value: "Search", direct:
    "true"])
}
```

然后使用下面这个测试方法：

```
public void searchIssue(String type, String issue){
    select "issueSearch.issueType", type
    keyType "issueSearch.searchBox", issue
    click "issueSearch.searchButton"
    waitForPageToLoad 30000
}
```

如果有一天，你需要把 Selector 修改成输入框，那我们只需要更新对应的 UI module：

```
ui.Form(uid: "issueSearch", clocator: [action: "list", method: "GET"]) {
    InputBox(uid: "issueType", clocator: [name: "can", direct: "true"])
    TextBox(uid: "searchLabel", clocator: [tag: "span", text: "for"])
    InputBox(uid: "searchBox", clocator: [type: "text", name: "q", id: "q"])
    SubmitButton(uid: "searchButton", clocator: [value: "Search", direct:
"true"])
}
```

然后修改命令：

```
select "issueSearch.issueType", type
```

为：

```
type "issueSearch.issueType", type
```

其余则保持不变。

如果有动态的 Web 内容，比如 Google Books 的网站，它包含了一个图书分类的列表，每个分类中包含一个图书列表。针对这样 UI 的 UI module 会出奇的简单：

```
ui.Container(uid: "GoogleBooksList", clocator: [tag: "table", id: "hp_table"])
{
    List(uid: "subcategory", clocator: [tag: "td", class: "sidebar"], separator:
    "div") {
    Container(uid: "{all}") {
        TextBox(uid: "title", clocator: [tag: "div", class: "sub_cat_title"])
        List(uid: "links", separator: "p") {
            UrlLink(uid: "{all}", clocator: [:])
                }
        }
        }}
```

Tellurium UID 描述语言为定义动态 Web 内容提供了更多的灵活性。我们来看个复杂点的例子。

```
ui.StandardTable(uid: "GT", clocator: [id: "xyz"], ht: "tbody"){
    TextBox(uid: "{header: first} as One", clocator: [tag: "th", text: "one"],
self:
    true)
    TextBox(uid: "{header: 2} as Two", clocator: [tag: "th", text: "two"], self:
true)
    TextBox(uid: "{header: last} as Three", clocator: [tag: "th", text: "three"],
    self: true)
    TextBox(uid: "{row: 1, column -> One} as A", clocator: [tag: "div", class:
"abc"])
    Container(uid: "{row: 1, column -> Two} as B"){
        InputBox(uid: "Input", clocator: [tag: "input", class: "123"])
        Container(uid: "Some", clocator: [tag: "div", class: "someclass"]){
            Span(uid: "Span", clocator: [tag: "span", class: "x"])
            UrlLink(uid: "Link", clocator: [:])
            }
           }
        TextBox(uid: "{row: 1, column -> Three} as Hello", clocator: [tag: "td"],
```

```
self:
       true)
   }
```

在这个例子中，我们是使用元数据“first”、数字和“last”来指定 header 的位置。元数据“{row: 1, column -> One} as A”意味着我们例子中的 UI 元素——一个 TextBox，处于第一行，并和 header“One”处在同一列。测试代码很干净，比如：

```
getText "GT.A"
keyType "GT.B.Input", input
click "GT.B.Some.Link"
waitForPageToLoad 30000
```

未来计划

Tellurium 是一个年轻而且创新的项目，它具有许多来自开发团队和用户社区的新颖思想。Tellurium 希望在以下领域继续发展：

1. Tellurium 0.7.0 已经实现了一个新的使用 jQuery 的测试驱动引擎。这个引擎的主要特性包括：UI module 组定位、UI module 缓存、命令集处理、用 jQuery 重新实现的 Selenium API 以及新的 Tellurium API。Tellurium 会继续开发这个新引擎直至完备。
2. Tellurium IDE module 插件 0.8.0 RC1 刚刚发布，它包含了很多新的特性。录制和生成测试脚本的 Tellurium IDE 的 RC 版本也已发布。它们是 Tellurium 成功的关键，因此会得到持续的改进。除了 Trump 和 Tellurium IDE，Tellurium 还在计划改善 TelluriumWorks，让它可以编辑、完成语法检查，并运行 Tellurium DSL 测试脚本。
3. Tellurium 作为云测试工具是另外一个非常重要的发展方向。项目团队正在计划重新考虑整个架构，使其可以更直接地并行执行测试。通过 P2P 技术，只需最少的管理，就让测试服务器在云的环境中具备自组织和自协调的能力，这是件极具挑战性的工作。

Tellurium 框架还希望发展的其他领域包括：

- 创建可重用的 Dojo、ExtJS 和 jQuery UI Tellurium widget。这样可以让其他人只需要在自己的项目中包含对应的 jar 文件，就可以重用这些 widget。
- 支持行为驱动测试。
- Web 安全测试
- 支持其他的语言，比如 Ruby

关于作者

方剑毕业于乔治亚理工学院，电子和计算机工程专业博士学位。他的工作是某 IT 公司的高级软件工程师，致力于设计和实现企业级应用软件。他是 Tellurium 自动化测试框架项目的创建者。

原文链接：http://www.infoq.com/cn/articles/tellurium_intro

相关内容：

- 开源自动测试框架Tellurium
- 测试覆盖率强迫症
- 书摘与采访：依赖注入
- FlexMonkey将单元测试引入Flex用户界面开发
- 自动化测试的分层结构

推荐文章 | Articles

Visual Studio 2010 中的扩展管理器

作者 Yiyi Sun 译者 侯伯薇

简介

扩展能够为核心应用程序提供附加的功能或者自定义设置。在很多应用程序中这都是很流行的，包括流行的 web 浏览器 IE 和 Firefox、微软的 Office、媒体播放器、Photoshop、Eclipse、Visual Studio 等等。其中有些可能比较新奇，而针对开发工具——像 Eclipse 和 Visual Studio——的扩展让开发者的生产力得到了显著提高。

Visual Studio 早期的版本是构建在 COM 技术之上的。 而 Visual Studio 的主要扩展方式插件是通过 COM 接口实现的。随着时间的变化，插件机制已经得到了改善。在 2002 年，随着 Visual Studio .NET 的发布，微软引入了.NET 和 COM 之间的互操作性。在 Visual Studio 2005 中，插件支持使用 XML 的简化部署方式。之后不久，微软发布了 Visual Studio SDK 开发包，这是一种更为新式的机制，通过底层的 API 提供了与 Visual Studio 的 IDE 更深层次的整合。

随着时间的推移，Visual Studio 扩展的部署技术也得到了改善。在 Visual Studio 2005 中引入了 Visual Studio 内容安装器（Content Installer VSI），它可以用于部署宏、插件、工具箱控件、代码片段以及模板。 而更广泛地用于应用程序安装的 Windows 安装包（MSI），则成为针对 Visual Studio 包的主要部署技术。VSI 和 MSI 格式的文件中都包含了完整的分发包。

这种设计导致了一些交付问题。为了获得扩展，用户需要在不计其数的网络站点中搜索、下载和安装。管理扩展的依赖关系和更新既不简单，也不易懂。

在 Visual Studio 2010 增加的多种新特性中，新的扩展管理器就是为了解决扩展部署问题的。它遵循了一种新的用于发布和部署第三方扩展的趋势，这种趋势是由于像 Firefox 和 Eclipse 这样的项目而变得流行。

本文会向你介绍 Visual Studio 2010 的扩展管理器。你会了解到如何构建简单的扩展，并把它部署到 Visual Studio Gallery 中。

崭新的用户体验

应用程序扩展是与核心应用程序分别发布的。传统上认为它们更像是独立的应用程序。 尽管随着时间的推移，Visual Studio 扩展性的很多方面都得到了改善，然而交付机制还是没有太多的改变。它遵循的是双击工作流（double-clicking workflow），那并不容易。

1. 在哪里能够找到扩展？
2. 如何管理扩展的生命周期，包括安装、更新和卸载？

在过去，想要找到针对 Visual Studio 的正确的扩展，我们需要做大量的搜索工作。开发者可以在 Internet 上搜索扩展，或者可以从杂志的文章中、论坛的帖子中或者博文中获得扩展的信息。但我们很难发现所有可用的选择，而且也很难对类似的扩展进行比较。

有了 Visual Studio Gallery 之后，这种状况已经得到了改变。这个站点是 Visual Studio 扩展的官方和集中资源所在。你可以浏览并且订阅扩展的 RSS。你还可以阅读评论，并提供你的反馈，对扩展打分。

Visual Studio 2010 的扩展管理器是与 Visual Studio Gallery 结合使用的。它把 Visual Studio Gallery 带到了 Visual Studio IDE 中（如图 1 所示）。我们不再需要使用 web 浏览器来搜索或者运行安装程序了。在 Visual Studio 2010 中，我们可以通过 Tools | Extension Manager 菜单来打开扩展管理器。当扩展管理器打开之后，它会列出已经安装了的扩展。如果对于扩展有更新可用，也会在其中列出来。在左边的面板中，点击“Online Gallery”标签页，你就可以浏览和搜索在“Visual Studio Gallery”中可用的扩展了。当你选择一项扩展，相关的描述就会显示在右侧的面板中。点击“下载”按钮，扩展管理器马上就会开始下载选中的扩展。

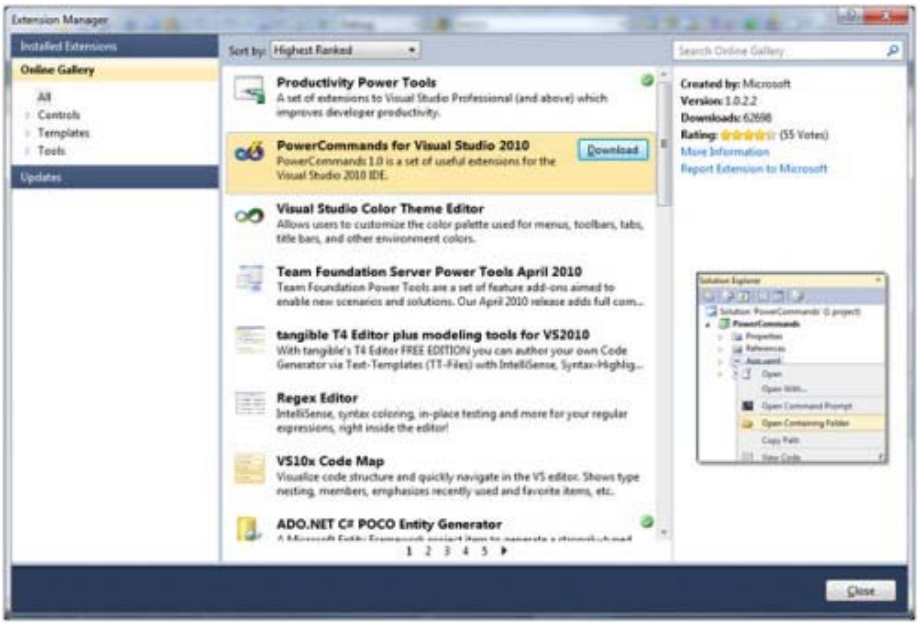


图 **1 Visual Studio 2010** 的扩展管理器

有时，安装一项扩展需要重启 Visual Studio。这也没关系。IDE 会返回之前的样子，载入最新的解决方案。删除扩展同样很容易。通过 Tools | Extension Manager 菜单打开扩展管理器，选择扩展，然后点击“Uninstall”按钮。

它不仅提供了简单的方式，让我们可以在 Visual Studio IDE 中搜索和安装扩展。还提供了对扩展完整的生命周期管理。我们还可以在 Visual Studio IDE 中对扩展进行启用/禁用、更新和卸载。

在扩展管理器中启用和禁用扩展的能力是值得一提的。正如你将在本文的下一部分将会看到的，创建扩展并把它上传到 Visual Studio Gallery 中要比之前容易得多。再也没有批准流程。上传的内容马上就可以访问。这可能有些风险。某些扩展会有质量问题，或者与其它扩展冲突。在那种情况下，禁用特性会为其提供帮助。

自动的扩展更新通知也是很棒的特性。每次你打开 VS 2010 的一个实例的时候，它就会自动检查已经安装的扩展的更新。当找到更新的时候，它就会在系统托盘中显示通知图标。点击通知，它就会载入扩展管理器来安装新的更新。这确保你能够拥有最新的扩展更新，而不需要手动地搜索更新。

有一些扩展对于开发者是必需的。在 Internet 上，人们会告诉你他们最喜欢的扩展。在此我列出了我所选择的部分。建议你尝试使用它们。那很容易，并且如果你不喜欢，总是可以禁用并卸载它们。

- **Visual Studio Productivity Power Tool**，这是一组用于提高生产力的工具，像解决方案导

航器、带有改良的标签页的用户界面、可搜索的添加引用对话框，等等。

- **Snippet Designer**，这种扩展让你可以在 Visual Studio 中创建代码片段。
- **Spell Check**，这种扩展为纯文本文件、注释、字符串和 HTML/ASP 代码提供了拼写检查功能。
- **Tangible T4 Editor**，这种扩展为 T4 模板提供了智能提示和语法的突出显示。
- **GhostDoc**，这是一种能够为 C#生成 XML 文档注释的扩展。
- **Feature Builder Power Tool**，这种扩展会帮助你创建扩展。
- **PowerGUI**，这种扩展为 PowerShell 脚本提供了智能提示和语法的突出显示。
- **IronPython IronRuby Console**，这种扩展让你能够在 IronPython/IronRuby 中与 Visual Studio 交互。

扩展管理器的体验让我们想到了一些直观的流程，像下载软件到 iPhone、下载插件到 Firefox、以及从 Amazon 下载书籍到 Kindle 上等等。这可能就是消费软件部署的趋势。它是基于 Internet 或者云，并且与应用程序整合的部署方式。根据最近在网上泄露的“Windows Store for Windows 8”，我们可能会在下一版本的 Windows 中看到这种软件部署方式。

现在，Visual Studio 2010 扩展管理器和 Visual Studio Gallery 更加以扩展为中心。将来，它会更加以用户为中心。我希望它会记住我所选择的扩展，并在所有我登陆的计算机中应用那些扩展。

在“创建简单的扩展”部分中，我会一步一步地创建 Hello World 扩展。

在尝试这些步骤之前，你需要确保已经下载并安装了 Visual Studio SDK。

创建 Visual Studio Package 项目

首先，使用“Other Project Types | Extensibilities”下面的 Visual Studio Package Project 模板创建新的项目。

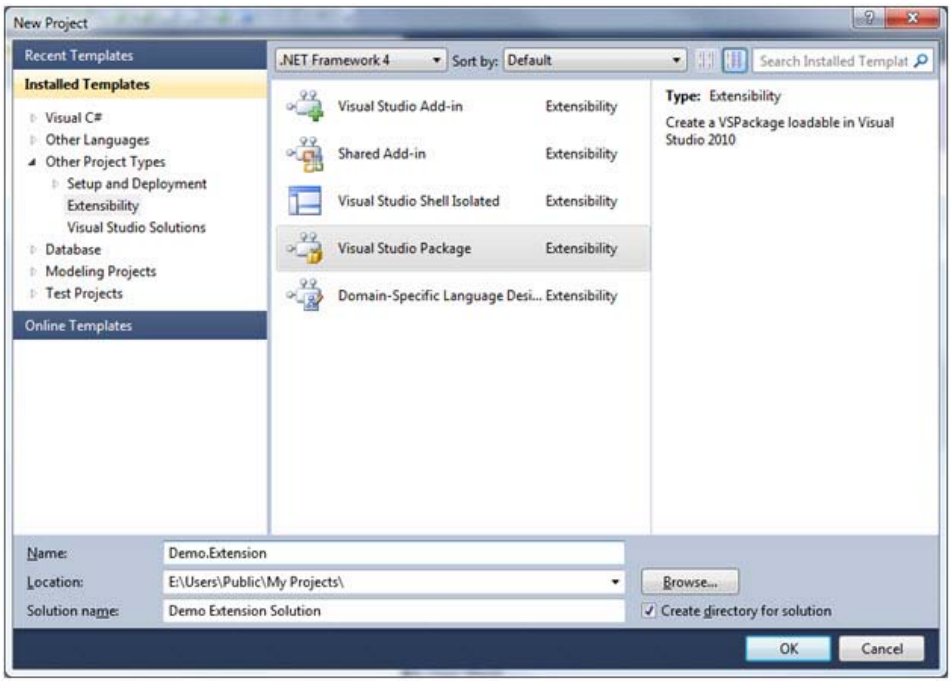


图 **2 Visual Studio Package Project** 模板

它会载入 Visual Studio Package 向导，它有七页，我们可以在其中为 Hello World 包填写基本的信息。这是很简单的过程。在第三步中，我选择了“Menu Command”和“Tool Window”选项，从而展现最初针对菜单项和工具窗口的代码。（图 3）

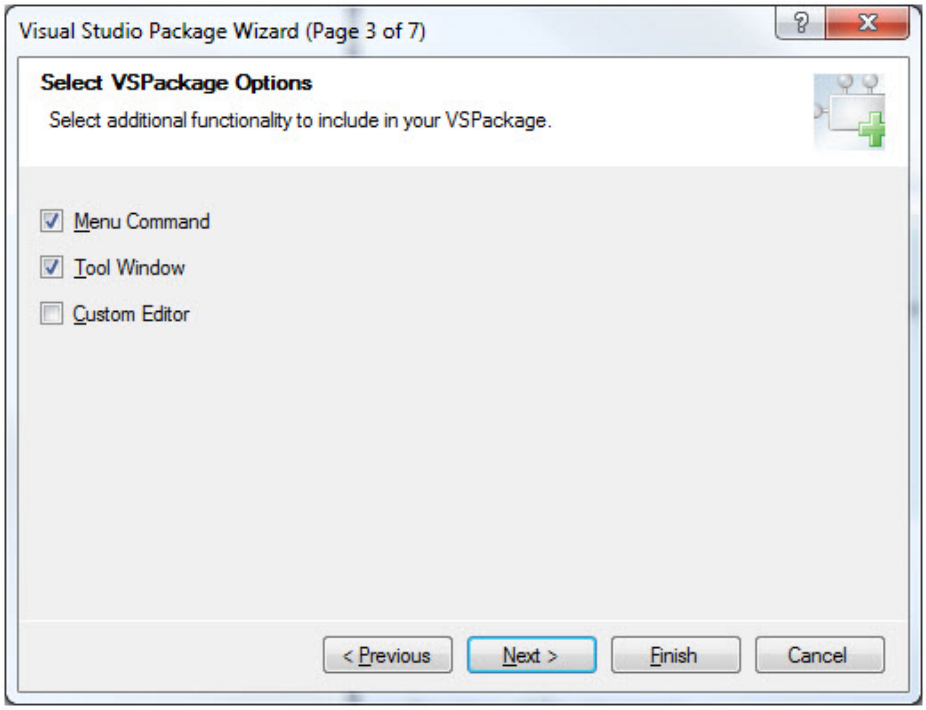


图 **3** 创建菜单命令和工具窗口

检查项目结构

在创建了项目之后，让我们来检查其中都有什么。有些东西是值得注意和说明的。

其中有一个 source.extension.vsixmanifest 文件。 在这个文件中，会生成我们在新建项目向导中输入的信息，还有包的引用和内容。它拥有属于自己的编辑器（图 4）。

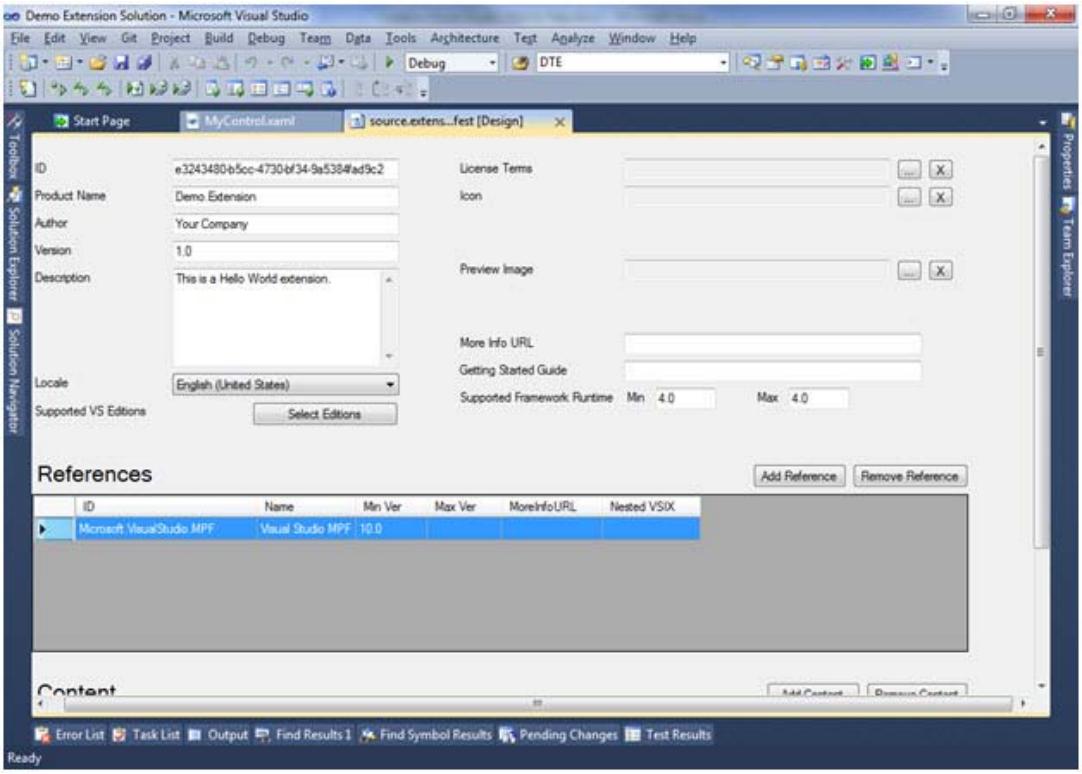


图 4 编辑器中的 **source.extension.vsixmanifest** 文件

在项目的属性页中，有一个新的 VSIX 标签页，让 Visual Studio 在调试扩展的时候，构建和/或部署 VSIX 的过程中创建 VSIX 文件。

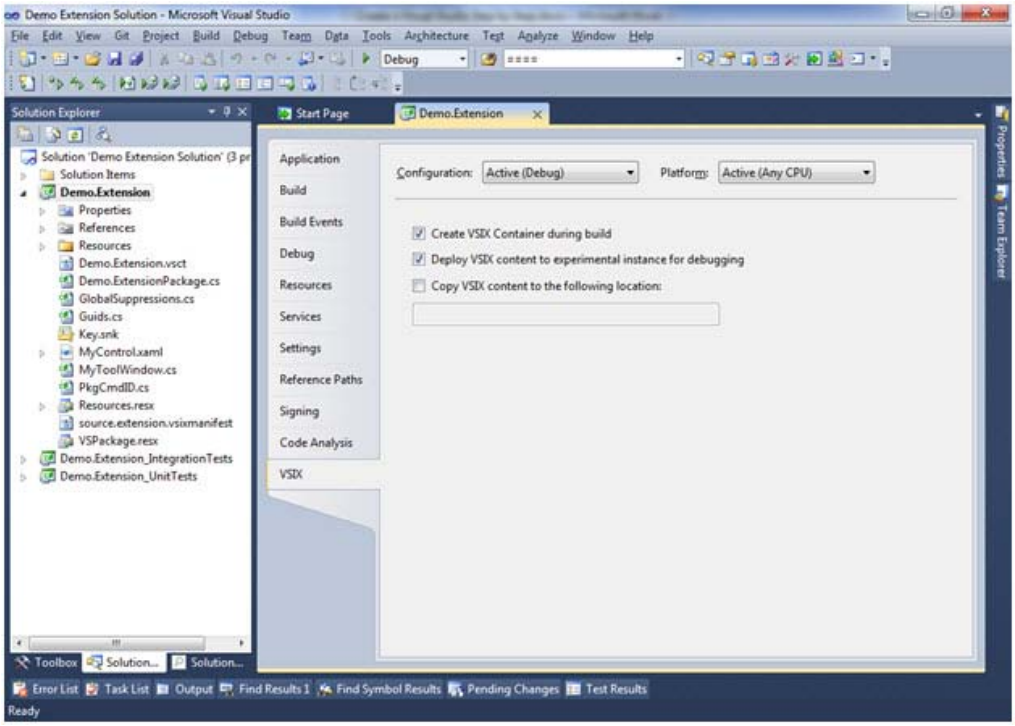


图 **4.1** 项目属性页中的 **VSIX** 标签页

在 Visual Studio 2010 之前，扩展是作为安装包部署的，这多少和单独的应用程序有些类似。它们是在 Visual Studio 之外使用 VSI 或者 MSI 分别安装和维护的。开发者经常会认为编写 Visual Studio 扩展很难，或者那不是一般开发者所能够完成的。那是对的。此外，为 Visual Studio 扩展创建安装包更困难。

为了给 Visual Studio 2008 扩展创建安装包，MSDN 上的一篇文章“教程： 简单的 VSPackage 部署”建议使用 xcopy、安装项目，或者使用 Windows Installer XML Toolset (WiX)。在所有情况下，开发者都不得不学习 MSI、WiX，或者理解注册表的键值以注册扩展。

使用安装项目的方法，开发者需要编写自定义的安装动作来注册扩展。使用 WiX 既耗时又容易出错，因为那需要手动编辑 XML 文件，并且维护很多 GUID 值。默认情况下，WiX 安装只有最少的功能。开发者需要操作更多的 XML 文件来提供功能，像定义安装文件夹以及显示许可信息等等。

在 Visual Studio 2010 中，对于开发扩展至少在两个方面做出了改进。使用 VSIX 来对扩展打包更容易。 VSIX 是在 Visual Studio 2010 中引入的针对扩展的部署单元。VSIX 文件是一种使用了开放打包协议（Open Packaging Convention）的压缩文件。我们可以从本地硬盘中像 VSI 和 MSI 的方式安装 VSIX。 Visual Studio 会识别出 VSIX 的文件扩展名，把它解压并把文件安装在正确的位置中。它还可以通过 Visual Studio Gallery 进行发布和升级。当你构建扩展项目的时候，Visual Studio 2010 扩展项目模板就会生成 VSIX 文件。

在 Visual Studio 2010 中，开发者需要处理 XML 文件。但是开发者不需要编辑原始的 XML 文件，Visual Studio 2010 为其提供了编辑器。那就是新的 VSIX 说明文件编辑器（图 3）。你所需要做的就是指定相同的 VSIX ID——唯一的标识符，以及新的版本号。然后把它上传到 Gallery 中，这样拥有旧版本的用户就可以下载更新。

VSIX 本质上是一种简单的 xcopy 部署方式。安装就是要对包进行复制和解压。卸载只是删除文件夹。它不支持高级的安装特性，像为 GAC 添加程序集或者创建桌面快捷方式等等。它也不会在 Visual Studio 之外写入注册表。它不支持自定义的动作。MSDN 上名为"Visual Studio 扩展部署"的文章对 VSIX 和 MSI 能做的和不能做的做了详细的说明。

VSIX 主要是为了 Visual Studio 扩展部署而设计的，并且使用 Visual Studio Gallery 来发布。

构建项目

在构建了项目之后，构建的输出包括封装后的 DLL、包定义文件（pkgdef)以及 VSIX 文件（图 5）。

VSIX 文件是包的部署文件，你可以把它发送给用户来进行安装，或者上传到 Visual Studio Gallery 中。

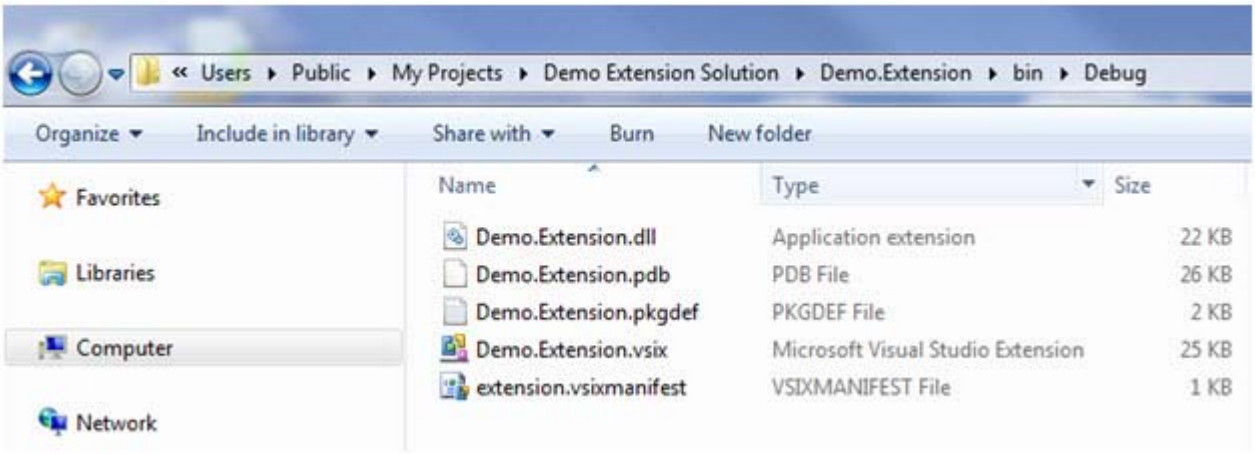


图 5 构建输出

调试和运行扩展

在项目的属性页中，调试选项被设置为载入 Visual Studio 实验环境。（图 6）

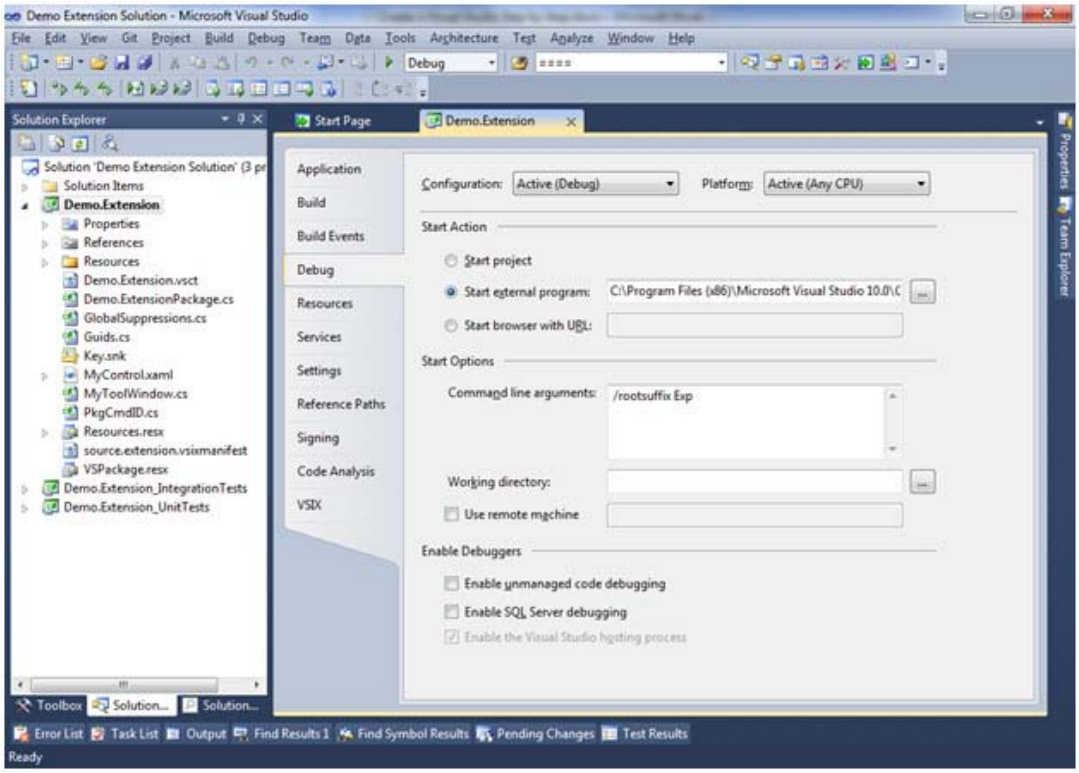


图 6 项目的调试设定

按下 F5，就会载入部署了 VSIX 的 Visual Studio 实验环境的新实例。Visual Studio 实验环境会与 Visual Studio 开发环境并行运行。它是用于开发和测试 Visual Studio 包的沙盒环境。部署在实验环境中的包不会影响你的主开发环境。实验环境拥有自己的设定、注册表项和扩展。我们可以使用开始菜单中的“重置微软 Visual Studio 2010 实验实例”选项，把它重置为干净的状态。

我们创建的简单扩展会在 Tools 菜单下添加“Hello World”菜单项（图 7），并在“View | Other Windows”下添加“Hello World Tool Window”菜单（图 8），我们可以使用它来显示工具窗口（图 9）。

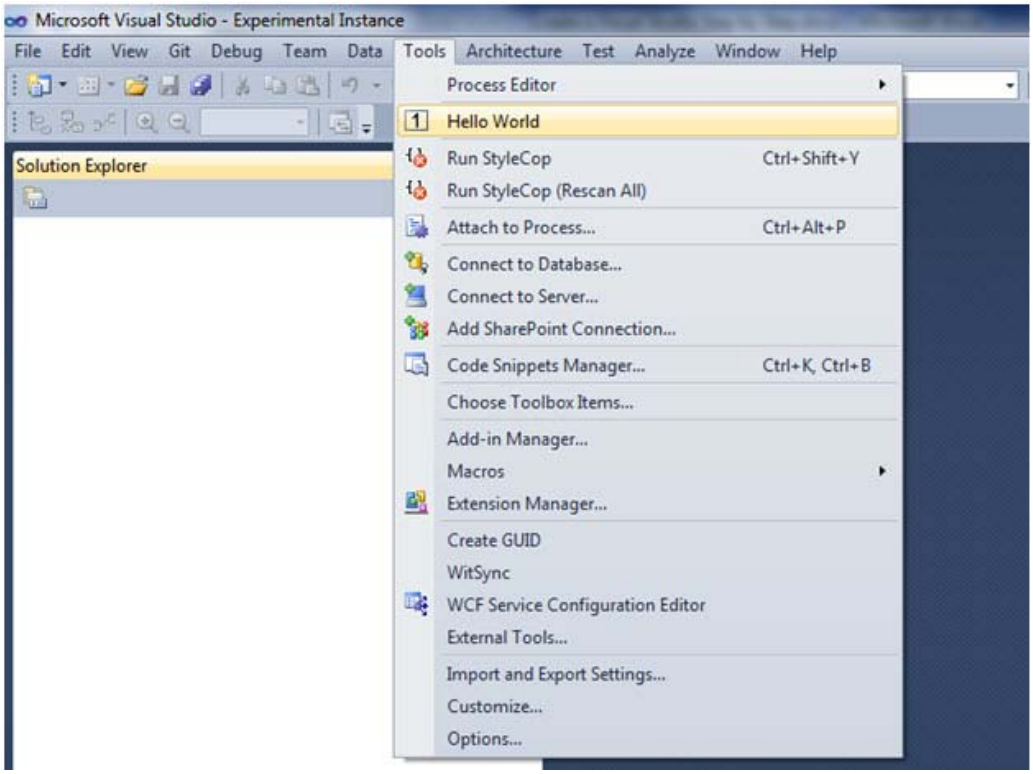


图 7 Toos 菜单下的“Hello World”菜单

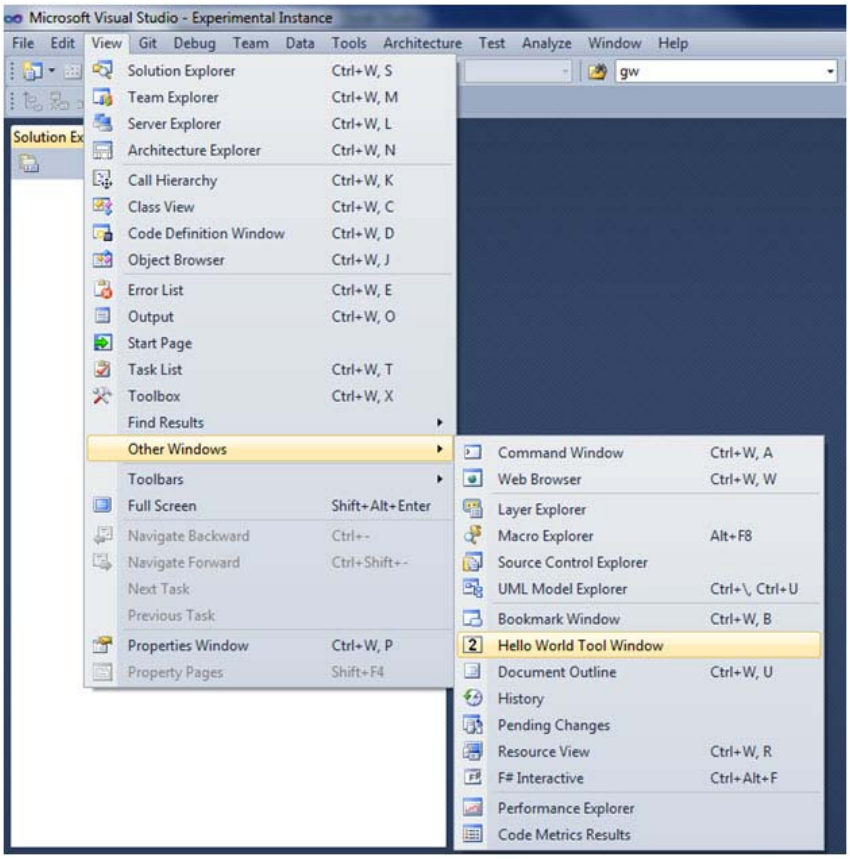


图 8 “View | Other Windows”下的“Hello World Tool Window”菜单

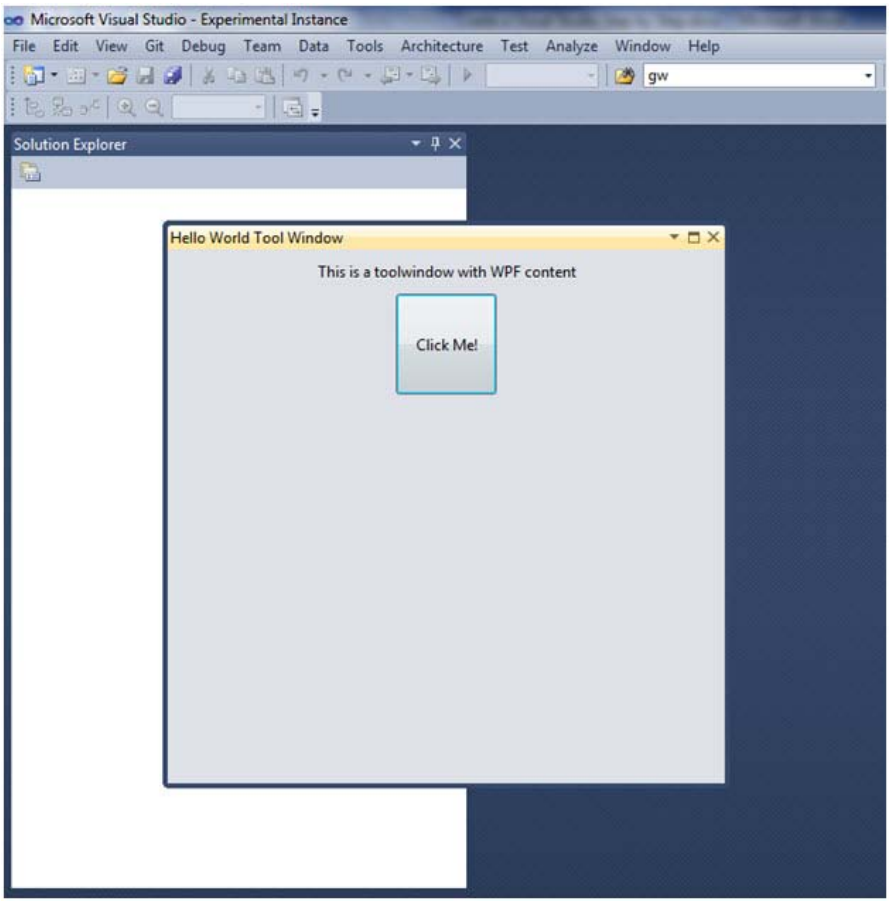


图 **9** 扩展的工具窗口

包的主要入口点在于 Microsoft.VisualStudio.Shell.Package 的子类中。它拥有多个与包的注册相关的属性，我们会在安装和注册的过程中使用它们。（图 10）

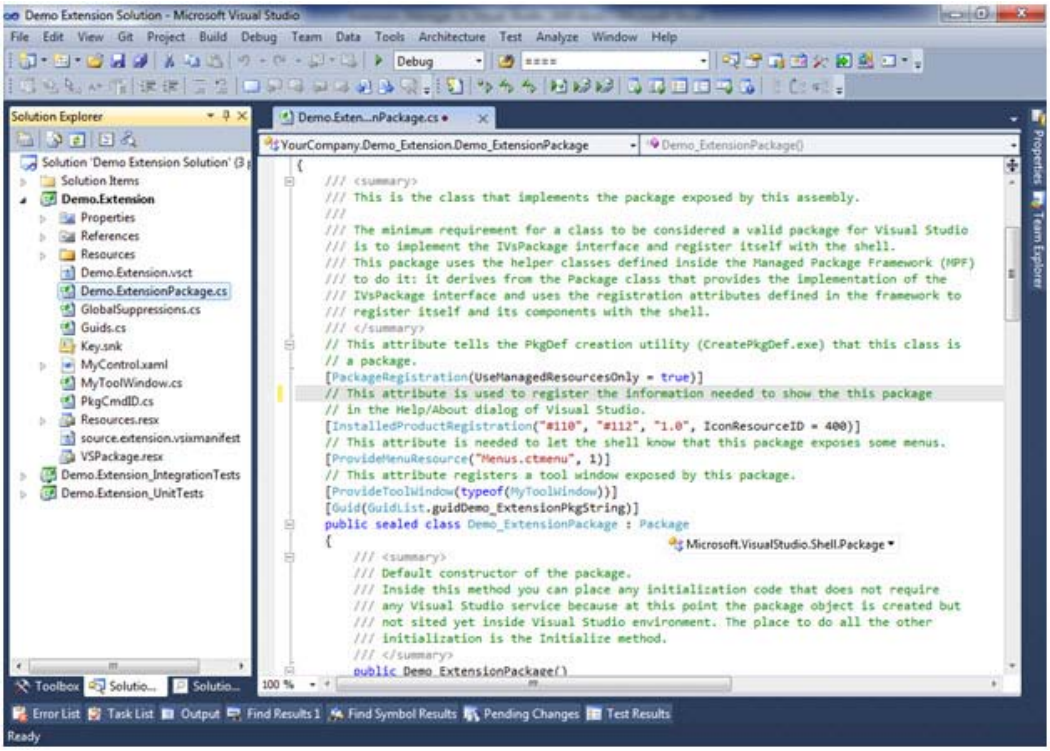


图 **10 Package** 类

你可以在名为 Initialize 的重载函数中编写代码，从而在载入你的包时与 Visual Studio 交互。在我们简单的包中，“Hello World”和“Hello World Tool Window”菜单都是在这个初始化函数中创建的 。

```
/////////////////////////////////////////////////////////////////////////////
// Overridden Package Implementation
#region Package Members

/// <summary>
/// Initialization of the package; this method is called right after the package is sited, so this is the place
/// where you can put all the initialization code that rely on services provided by VisualStudio.
/// </summary>
protected override void Initialize()
{
    Trace.WriteLine (string.Format(CultureInfo.CurrentCulture, "Entering Initialize() of: {0}", this.ToString()));
    base.Initialize();

    // Add our command handlers for menu (commands must exist in the .vsct file)
    OleMenuCommandService mcs = GetService(typeof(IMenuCommandService)) as OleMenuCommandService;
    if ( null != mcs )
    {
        // Create the command for the menu item.
        CommandID menuCommandID = new CommandID(GuidList.guidDemo_ExtensionCmdSet, (int)PkgCmdIDList.cmdidHelloWorld);
        MenuCommand menuItem = new MenuCommand(MenuItemCallback, menuCommandID );
        mcs.AddCommand( menuItem );
        // Create the command for the tool window
        CommandID toolwndCommandID = new CommandID(GuidList.guidDemo_ExtensionCmdSet, (int)PkgCmdIDList.cmdidHelloWorldTool);
        MenuCommand menuToolWin = new MenuCommand(ShowToolWindow, toolwndCommandID);
        mcs.AddCommand( menuToolWin );
    }
}
#endregion
```

图 **11 Initialize** 函数

菜单的回调函数是 MenuItemCallBack 和 ShowToolWindow。当点击“Hello World”菜单的时候，就会触发 MenuItemCallBack 函数。默认的代码是要显示一个消息框。它是你的包逻辑

的起始点。当点击“Hello World Tool Window”菜单的时候，就会触发 ShowToolWindow 函数。默认的代码是以标准的方式弹出工具窗口。

工具窗口是 Visual Studio IDE 的子窗口。和工具箱、解决方案管理器和属性窗口一样，它可以停靠、浮动或者以标签的形式显示在文档框架中。它们是动态的（或者说是自动可见的）。有了 Visual Studio 2010 对多显示器的支持，工具窗口也可以脱离开来，并拖拽到第二台显示器中。Visual Studio Package Project 模板默认情况下会为工具窗口创建 WPF 的用户控件。

现在我们已经有了 Visual Studio Package 的骨架，可以构建他来生成可以分发的 VSIX 文件，我们或者可以通过复制的形式或者通过 Visual Studio Gallery 来分发它。

发布到 Visual Studio Gallery

想要通过 Visual Studio Gallery 来发布扩展，只需要三步，就可以发布你的扩展，从而让全世界的开发者都能够访问到它。

- 选择你的扩展的类型
- 上传 VSIX 文件
- 添加附加的类别和描述。

Visual Studio Gallery 会从 VSIX 中取得元数据，包括名称、ID、版本、公司、许可以及产品的图片。扩展管理器会使用版本号来自动管理扩展的更新。

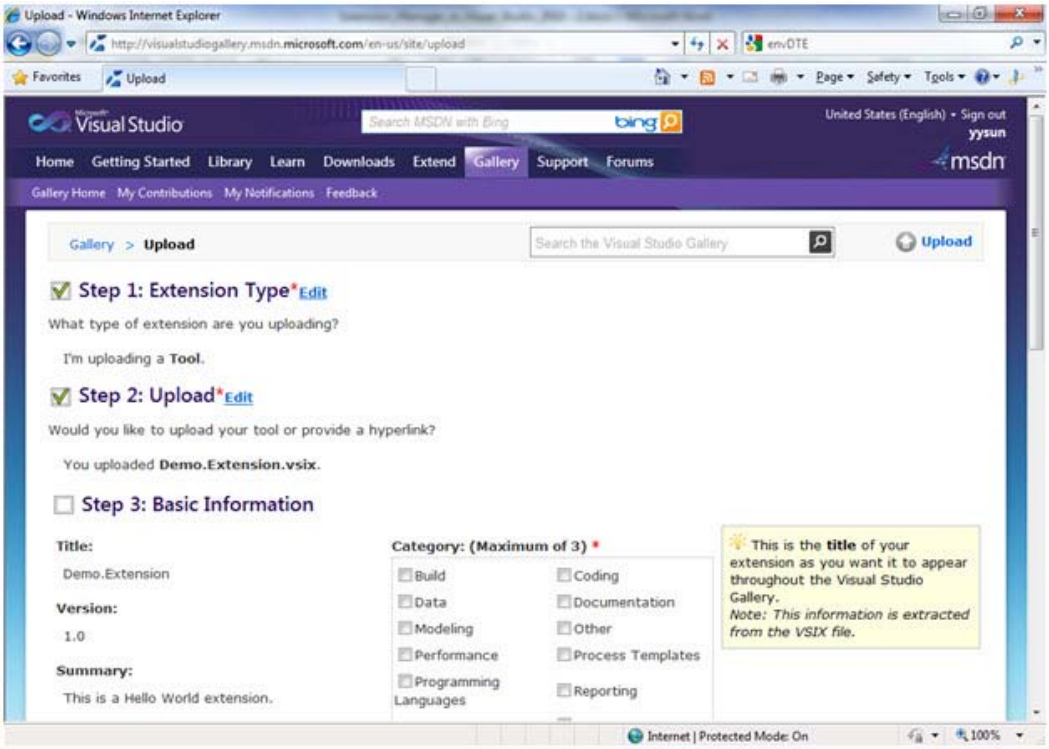


图 **12 Visual Studio Gallery**

当你的扩展有新版本时，你所需要做的就是在 VSIX 的说明文件编辑器中更改版本号，构建新的 VSIX 文件并上传到 Visual Studio Gallery 中。这样，更新就马上可以提供给用户下载了。当用户启动 Visual Studio 新的实例时，Visual Studio 就会在系统托盘显示新的更新通知。

有时这是很方便的，因为 Visual Studio Gallery 没有验证过程。发布扩展就像在推特上发布一条消息一样简单。对扩展开发者的一项提示是，你应该尽可能地测试以保证质量。你不应该先上传扩展，然后才开始测试，因为它马上会被大家所发现。VSIX 支持双击安装，就像 VSI 和 MSI 一样。如果其中有 bug，那么就使用扩展管理器来卸载它，创建新构建的 VSIX，然后双击它来安装并进行测试。在开发过程中，你不需要提升版本号。记着在完整地测试之后在把它上传到 Visual Studio Gallery 中。

这会带来另一种提示。 Visual Studio Gallery 不支持私有分发。如果你的扩展只在你的公司的开发者中使用，那么你就不应该把它上传到 Visual Studio Gallery 中。相反，你应该在公司内部发布 VSIX 文件。在这种情况下，想要更新扩展，你需要手动地与扩展用户沟通。

Visual Studio Gallery 的目的之一就是要创建 Visual Studio 扩展的生态系统。自从几年之前它可以用于 Visual Studio 2008 开始，人们已经上传了上千个扩展。与其说是一种商业集市，不如说是一种社区分享站点，就像 Apple 的 App Store 和 Windows Phone Marketplace 一样。如果开发者想要销售他们的扩展，那么就需要找到其它厂商和付费服务，就像旧式的共享软件

发布一样。 用户无法通过扩展管理器付费。结果是，扩展管理器和 Visual Studio Gallery 为开源和免费的扩展项目提供了优秀的支持。

总结

Visual Studio 2010 的扩展管理器和 Visual Studio Gallery 提升了扩展用户和扩展开发者的用户体验。你应该始终记着，那里有成千上万的扩展可供你扩充 Visual Studio 的特性。还应该始终记着 ,你可以在 Visual Studio 中搜索和管理扩展的生命周期。如果你对于扩展 Visual Studio 有什么好主意，那么我建议你试着创建一些 Visual Studio 扩展。Visual Studio 扩展会提升开发的生产力。

原文链接：http://www.infoq.com/cn/articles/extension-manager-2010

相关内容：

- 图书聚焦：Visual Studio 2008 揭秘
- Dan Lucraft谈NoSQL DB在Rails中的使用与基于JRuby的编辑器RedCar的设计思路
- Word自动化局部架构设计
- Visual Studio .NET使用技巧手册
- 基于Windows Azure的云计算应用设计

推荐文章 | Articles

淘宝 UED 技术负责人小马畅谈 Web 前端开发

作者 崔康

小马（赵泽欣）是淘宝 UED 团队的技术负责人，在 Web 开发领域有丰富的经验。InfoQ 编辑在 QCon 全球企业开发大会（北京站）期间，就 Web 框架选取标准、性能优化建议、前端工程师角色定位等相关问题对他作了专访。

InfoQ：小马，请首先做一下自我介绍。

小马：大家好，我叫赵泽欣，在淘宝的花名是小马，隶属于淘宝 UED 部门，很荣幸能够接受 InfoQ 的采访，谢谢。

InfoQ：小马，现在 **Web** 开发相关的类库特别多，请问淘宝在选择 **Web** 框架做开发的时候，是从哪些角度来考虑的？能否给读者朋友提供一些参考？

小马：的确如你所说，Web 类库五花八门。我们在做选择的时候，主要基于以下这两个方面：

1. 想想是否真正符合自己的需求，有些类库可能适合网站型应用，另一些可能适合数据型应用，所以其实还是有些区别的，这是第一点。
2. 我觉得还要看人才或者是技术储备的状况。如果你的团队里面没有几个特别熟悉这种类库的人才，有时候会吃一些苦头。

像淘宝网当初选择 YUI 也是基于这两个因素考虑。因为淘宝的后端架构是基于 JAVA 的，而 YUI 是大多数框架里面，在文档组织、代码组织上最接近于 JAVA 的，比较容易得到开发人员的认可，然后功能也比较强大，组件比较丰富，再加上又有一个专业的雅虎团队在维护，所以可靠性上也会比较强，因此当时选择它。最重要的还是根据你自己的需求。

InfoQ：目前淘宝 **UED** 团队已经发展壮大，你们现在是基于 **YUI** 呢，还是有进一步的计划？

小马：目前大部分还是基于 YUI，因为已经用了很久了，也基于它做了很多改进和扩展。但是随着淘宝飞速的发展，各种应用也越来越复杂，我们 也慢慢感受到 YUI 的一些不足，YUI

当然它自己也在发展，推出 YUI 3 的版本，也做了很多的改进。可是我们在用的过程中，觉得还是不太适合，不太适用于淘宝，所以我们也有计划的在研发自己的框架，我们的框架叫做 Kissy，我的同事玉伯主持这个项目，我们把它开源了，我们希望国内外更多的朋友能够一起来丰富这个框架。

InfoQ：我们知道前端很注重用户体验，比如响应时间等，现在有关 **Web** 性能优化的规范特别特多，我想问一下，以你的经验来说，有哪一些建议比较重要？

小马：当初我们做性能优化，最主要还是基于雅虎的 YSlow 工具，他们专门有一个性能团队，做了很多的研究，列出了很多优化的原则，我们刚开始还是拿来主义，学习他们，所以有几条我觉得特别能够立竿见影的方法。第一是把很多的脚本文件合并成一个文件，或者是把很多的 CSS 文件合并成一个文件，这样能够有效的减少 HTTP 的请求。然后尽可能把脚本放在页面靠结束的部分，把 CSS 尽可能放在靠前的部分。以我的经验，这些措施能够很快让用户感受到速度的一个提升。如果要做更多的优化的话，那可以按照 YSlow 的规则去做，其实性能优化不光是前端的职责，有时候也需要后端的支持，比如说 Cache，有时候是需要 Apache 或者 http server 的一些配置，其实性能需要整个公司里面各个团队一起来配合。

InfoQ：前端团队除了注重性能之外，是否还有其他领域，比如安全性，淘宝是不是也很注重这方面的问题？

小马：我个人觉得其实安全性也是一个我们很关注的地方，因为现在很多安全性问题，可能都是以脚本作为跳板，很多也是基于浏览器来做一些攻击。前端最熟悉浏览器，也是对脚本最为熟悉的一个岗位，所以很多时候，跟公司的安全技术部门交流的过程中，我们都希望从前端的角度给他们更多的建议。除了安全，我个人觉得，淘宝现在比较关注一些手持设备上面的前端研发，比如 iPhone、iPad 等。

InfoQ：淘宝 **UED** 团队目前关注哪些最新的 **Web** 前端技术？

小马：现在可能最热门的就是 HTML 5 和 CSS 3，但是很遗憾 IE6、IE7 这些浏览器还占据大量的市场份额，像淘宝基本上 90%的用户浏览器都还是 IE 系列。我们以前经常会有这种想法“反正用户都用 IE，这些新的东西用了也没什么意义”，现在我们慢慢的觉得，还是要推动这个事情，所以 HTML 5、CSS 3 的一些新特性，我们都尽可能的去应用，而且我觉得效果非常的好。

InfoQ：您作为淘宝前端团队的负责人，如何定位前端工程师这样一个角色？他的工作范围跟其他的工程师，比如后台开发工程师、还有一些专门做页面设计的工程师，有何区分？

小马：这个问题我们自己内部也有讨论过，我想了很久，提出一个定位，当然也是这个团队

大家一起的思考成果，我们觉得前端是面向用户的程序，首先还是程序，我们也要编码，我们也要具备严谨好学的特质，另外一点，我们跟很其他后端开发工程师有一些区别，就是可能我们更贴近用户，我个人觉得可能后端工程师更靠近机器，我们更靠近用户，所以我们需要更多的倾听用户的声音，所以我觉得前端工程师是面向用户的程序员。

InfoQ：目前云计算的实践越来越多，您觉得云计算会给前端工程师或者前端开发带来什么样的影响或者变化？

小马：现在很多人在谈这个概念，包括淘宝的测试团队，他们已经推出云测试工程师，所以我觉得前端工程师以后可能会慢慢的转化为云端工程师，因为在我的感觉里面，云其实就是一个大的存储的中心，一个计算服务中心，他能够提供计算的能力，能够提供存储，能够提供这些 SERVICE。那浏览器呢，又是一个比较重要的连接云端的工具，不管是浏览器还是基于浏览器的 OS，我觉得前端工程师具备很多技能，可以比较容易的转换到云端工程师，我自己是觉得这是一个很好的发展前景。

InfoQ：最后一个问题，我们 **InfoQ** 的读者，当中有很多一部分，可能是从事这种前端的开发，或者说以后有志于从事前端开发的，那么我想替他们问一个问题，因为您是淘宝的第一位前端工程师，可以说是经验非常丰富，您认为一个优秀的前端工程师，他应该具备哪些的素质，或者您在招聘一个员工的时候，会看中他哪些特质？

小马：那我直接说招聘好了，我一般会比较注重他的从不同角度去思考问题的能力。因为前端就是这样，很多时候处理一个问题，其实有多套的方案，关键是怎么去权衡，找到最合适于这个需求的一个方案，所以我特别看中他们会有这种从不同角度去看问题的这样一个能力；然后第二点，因为前端确实是一个新的岗位，很多新的东西，我还是比较看重他的学习能力；另外还有一点就是他真的是热爱前端的这样的一个岗位。

InfoQ：感谢小马接受采访。

小马：谢谢。

原文链接：http://www.infoq.com/cn/articles/zzx-web-development

相关内容：

- Blaze Data Services还是LiveCycle Data Services？
- 互联网开放平台技术趋势和讨论、Open API分析、实践和思索
- FlexMonkey将单元测试引入Flex用户界面开发

时刻关注企业软件开发领域的变化与创新

Java — .NET — Ruby — SOA — Agile — Architecture

Java社区：企业Java社区的变化与创新

.NET社区：.NET和微软的其它企业软件开发解决方案

Ruby社区：面向Web和企业开发的Ruby，主要关注Ruby on Rails

SOA社区：关于大中型企业内面向服务架构的一切

Agile社区：敏捷软件开发和项目经理

Architecture社区：设计、技术趋势及架构师所感兴趣的话题

讨 论 组：groups.google.com/group/infoqchina
编辑电邮：editors@cn.infoq.com
广告合作：sales@cn.infoq.com

新品推荐 | New Products

IBM 发布 WebSphere 应用服务器 V8.0 测试版

作者 Charles Humble 译者 王丽娟

IBM 最近公布了 WebSphere 应用服务器 V8.0 的测试计划，这一版本最终会支持 Java EE 6 规范。包括 WebSphere 8 在内，现在已经有七款产品即将对 Java EE 6 的两个 Profile 择一提供支持了。

原文链接：http://www.infoq.com/cn/news/2010/11/was_8

WSO2 发布 Carbon Studio

作者 Jean-Jacques Dubray 译者 吴宇

开源软件厂商，WSO2，针对其中间件平台于上个月发布了一套完整的基于 Eclipse 的开发环境。该工具简化了开发、测试以及部署服务、中介流和 google 小配件（gadgets）。

原文链接：http://www.infoq.com/cn/news/2010/11/carbon-studio

Eclipse Virgo 发布了

作者 Alex Blewitt 译者 张龙

近日，Eclipse Virgo 发布了，其中包含了 Eclipse Gemini Web 组件，虽然这是以 Eclipse 名义首次发布的版本，但它却是知名的 Spring DM Server 自然而然的演化。请阅读本文以了解关于此次发布的一些重要事项。

原文链接：http://www.infoq.com/cn/news/2010/11/eclipse-virgo-released

Google 发布了 mod_pagespeed

作者 Alex Blewitt 译者 侯伯薇

Google 最近发布了 mod_pagespeed，这是针对 Apache HTTPD 的模块，用来加速网页服务。它声称，通过添加 HTTP 缓存的友好消息，可以让速度加倍，而不需要修改服务器中的 web 应用程序。

原文链接：http://www.infoq.com/cn/news/2010/11/mod-pagespeed

Spring Python 1.2.0 RC1 发布了

作者 张龙

近日 ,SpringSource 发布了 Spring Python 1.2.0 RC1。Spring Python 借鉴了 Spring 框架的诸多概念并将其应用到了 Python 当中，这包括依赖注入、面向方面的编程、数据访问、事务、安全、远程服务以及 JMS 支持等。

原文链接：http://www.infoq.com/cn/news/2010/11/spring-python-1-2-rc1-release

Hive 0.6.0 发布了

作者 张龙

近日，Apache 软件基金会发布了 Hive 0.6.0。Hive 是由众多的志愿者所创建的开源项目。此前，它是 Hadoop 的一个子项目，但现在已经演化成为了 Apache 的一个顶级项目。

原文链接：http://www.infoq.com/cn/news/2010/11/hive-0.6-release

推荐编辑 | 翻译团队主编 张龙

我与 InfoQ 中文站的这几年

大家好，我是 InfoQ 中文站翻译团队的主编张龙，很高兴能在这期《架构师》专栏中再次与大家见面。

从 07 年底至今，我在 InfoQ 中文站已经度过了三个春秋，三年光阴弹指一挥间，自己也成为了 InfoQ 中文站发展的见证人。InfoQ 中文站作为国内在线媒体行业的后起之秀，以其高质量的品质、迅捷的反应速度和符合广大开发者、项目经理以及架构师的风格早已成为国内高端技术媒体的风向标。自己也在这三年中不断成长、成熟起来。现在的我已经养成了一个习惯：每天工作之前首先会打开 InfoQ，看看业界有何最新的消息以及技术的发展方向，可以这么说，InfoQ 中文站已经成为自己生活的一部分了。

InfoQ 中文站是个非常典型的分布式团队，团队成员来自大江南北：北京、成都、上海、西安、深圳、香港，甚至还有海外。如何管理好整个团队呢？其实根本无需过多的管理，一切管理的核心因素都是人。InfoQ 中文站的编辑完全是凭借自己的热情和想为社区贡献力量的心态加入的，这里没有高低贵贱之分，大家有如兄弟姐妹一样，互相交流心得，切磋技术。而我们的线下聚会则是增强团队凝聚力的最好方式，还记得 Jason 在聚会时的搞笑、泰稳打牌时的认真、李明一展歌喉时的帅气。每个人都为自己能加入这样的团队而感到自豪和兴奋。

这里的每个人都将为社区贡献力量视为己任，因为大家都是利用工作之余参与 InfoQ 内容的建设的。没有热情、没有坚持是撑不下去的。还好，我们一直在坚持，我们一直在努力，我们一直在"时刻关注企业软件开发领域的变化与创新"，这是我们前进的动力，这是我们力量的源泉。

随着 InfoQ 规模的不断壮大，我们还需要更多有理想，有奉献精神、愿意参与社区建设的朋友加入到我们的大家庭中。亲爱的朋友们 ,如果你想为社区的发展贡献自己的一份绵薄之力，如果你想体会到 InfoQ 中文站 Extremely Open 的管理与合作方式，如果你对业界的发展趋势很有想法，如果你有良好的文笔与技术功底，如果…，那么请加入到我们当中来吧，InfoQ 中文站欢迎每一位有想法、懂坚持的朋友。

封面植物

金花茶

金花茶，中国特产的传统名花，也是世界性的名贵观赏植物，我国一级保护植物。据统计，总数约有 220 种。而经自然杂交及人工培育的品种当在数千种以上。但以前，人们没有见到过花色金黄的种类。1960 年，中国科学工作者首次在广西南宁一带发现了一种金黄色的山茶花，被命名为金花茶。

生长分布：金花茶是一种古老的植物，极为罕见，分布极其狭窄，全世界 90%的野生金花茶仅分布于我国广西防城港市十万大山的兰山支脉一带，生长于海拔 700 米以下，以海拔 200-500 米之间的范围较常见，垂直分布的下限为海拔 20 米左右。

金花茶喜温暖湿润气候，喜欢排水良好的酸性土壤，苗期喜荫蔽，进入花期后，颇喜透射阳光。对土壤要求不严，微酸性至中性均土壤中可生长。耐瘠薄，也喜肥。耐涝力强。

植物形态：常绿灌木或小乔木，高 2-- 6m，树皮灰白色，平滑。叶互生，宽披针形至长椭圆形。花单生叶腋或近顶生，花金黄色，开放时呈杯状、壶状或碗状，径 3— 3．5cm；花瓣 9--11 枚，阔卵形至倒卵形 或矩圆形，肉质，具蜡质光泽；花期 11 月至翌年 3 月。蒴果三角状扁球形，黄绿色或紫褐色；果期 10-12 月。

用途：金花茶花色金黄，多数种具蜡质光泽，晶莹可爱，花型有杯状、壶状、碗状和盘状等，形态多样，秀丽雅致，在山茶 类群中，被誉为“茶族皇后”。亚热带地区可植于常绿阔叶树群下或植荫棚中，供以观赏。其花可入药。金花茶在民间一直被用于提神醒脑、清肝火、解热毒、养元气。

1kg.org 多背一公斤

爱自然 | 更爱孩子

架构师 12 月刊

每月 8 日出版

本期主编：崔康

总编辑：霍泰稳

编辑：李明 胡键 宋玮 郑柯 朱永光 池建强

读者反馈：editors@cn.infoq.com

投稿：editors@cn.infoq.com

交流群组：
http://groups.google.com/group/infoqchina

商务合作：sales@cn.infoq.com 13911020445

本期主编：崔康，**InfoQ** 中文站编辑

崔康，毕业于天津大学，计算机应用技术专业硕士，软件工程师，关注 Web 应用的开发与测试，感兴趣的领域包括 Web 性能诊断与调优等。希望能在工作之余，为中国软件技术的发展做一些力所能及的事情。联系方式为：cuikang[at]gmail.com。

InfoQ Enterprise Software Development Community

《架构师》月刊由 InfoQ 中文站出品。

所有内容版权均属 C4Media Inc.所有，未经许可不得转载。

时刻关注企业软件开发领域的变化与创新

架构师

www.infoq.com/cn/architect

每月8号出版

InfoQ neue
Enterprise Software Development Community

商务合作：sales@cn.infoq.com
读者反馈/投稿：editors@cn.infoq.com