Atitit 2012 技术趋势

目录

[1.1. 演进式架构（evolutionary architecture）来替代传统的预先设计出的重量级企业架构。 1](#_Toc13376)

[1.2. etty这样的嵌入式servlet容器（ Embedding a servlet container） 1](#_Toc5992)

[1.3. Windows环境自动化（Windows Infrastructure Automation）powerShell 2](#_Toc7468)

[1.4. 测试录制工具（Test Recorder） 2](#_Toc9497)

[1.5. 混合持久化（Polyglot persistence）是指根据效率和数据使用方式的差异而使用不同的数据存储方式的技术 3](#_Toc3287)

[1.6. 图形数据库Neo4j） 3](#_Toc13695)

[1.7. JavaScript微框架（Javascript micro frameworks 3](#_Toc16531)

[1.8. View层没有逻辑（logic-free markup）， Mustache 4](#_Toc25124)

[1.9. 应考虑使用Gradle替代Maven。 4](#_Toc32570)

[1.10. 领域特定语言（Domain Specific Language, DSL）声 5](#_Toc1036)

[1.11. 声明性配置（declarative provisioning） 5](#_Toc32012)

[1.12. Highcharts 互动图表可视化效果 5](#_Toc26776)

[1.13. MongoDB 5](#_Toc7039)

[1.14. JavaScript作为一个平台(JavaScript as a platform) 6](#_Toc30363)

[2. 不推荐 6](#_Toc19588)

## ****演进式架构****（evolutionary architecture）来替代传统的预先设计出的重量级企业架构。

## etty这样的****嵌入式servlet容器****（ Embedding a servlet container）

要比让应用运行在一个容器中有更多的好处。由于应用的启动过程更简单、开发环境更接近于生产环境，因此测试变的相对容易。因为多个应用无需共享库或驱动程序，所以不会出现库或驱动版本不匹配等令人讨厌的问题。虽然采用这种模型需要在生产环境下同时管理和监控多个Java虚拟机，但这种模型的简单性和隔离性所带来的优势更为显著。

## Windows环境自动化（Windows Infrastructure Automation）powerShell

owerShell等成熟的工具，搭配使用Chef和Puppet等新工具，是驱使我们本期《技术雷达》重点介绍****Windows环境自动化****（Windows Infrastructure Automation）的主要因素。使用鼠标和菜单选项手动配置环境太慢，而且在错误配置时易导致机器处于未知不可用状态。我们建议使用命令行工具来确保配置过程可脚本化，从而更加清晰。

## ****测试录制工具****（Test Recorder）

看似很有价值，因为它们提供了一种快速捕获录制用户对应用程序操作的方式。然而我们反对将测试录制工具作为一种常用工具，因为录制的测试往往非常脆弱，会因为UI的轻微变化而导致失败。测试录制工具产生的测试代码质量往往相对较差，经常出现不必要的重复。更重要的是，测试录制工具会阻碍测试自动化团队和开发团队之间的沟通。当面对一个很难通过用户界面进行测试的应用时，团队之间进行一次重要的谈话从而构建一个更容易测试的UI才是一个合理的解决方案

## ****混合持久化****（Polyglot persistence）是指根据效率和数据使用方式的差异而使用不同的数据存储方式的技术

。不是所有的应用场景都适合使用RDBMS做为默认的数据库。相反，先问这样的问题： session相关数据是否应该保存在这个数据库中？还是应该保存在某个Key-Value存储系统中？客户和产品的关系数据是否应该保存于图形数据库中？即使是在一个应用程序内，使用如MongoDB,Riak或Neo4J等NoSQL数据库，也会让我们重新考虑如何管理数据。

## 图形数据库Neo4j）

图形数据库将信息存储为由命名关系形成的任意互连节点，而不是表和连接。没有Schema，高度的可扩展性，图形数据库是复杂领域建模半结构化数据的一个非常好的选择。****Neo4j****（Neo4j）是该领域的领跑者，它的REST API和Cypher查询语言支持简单快速的存储和图表的遍历。

## ****JavaScript微框架****（Javascript micro frameworks

随着移动应用的兴起，JavaScript的文件大小和性能变得尤为重要。作为对一些大库的直接回应，****JavaScript微框架****（Javascript micro frameworks）出现了。这些轻量级的框架只做一件事情，比如DOM操作或MVC，并且大小在1KB内。通过一些微框架的组合使用，开发人员就可以获得他们想要的所有功能，并且不需要大库的开销。Microjs.com里有大量这样的微框架，当然也有工具可以将这些微框架捆绑成一个单独的库。

## ****View层没有逻辑****（logic-free markup）， Mustache

虽然过去几年中MVC一直是web开发的主流模式，但大多数库和框架未能遵守其最重要的原则之一：保证View层无逻辑。如果不能保证****View层没有逻辑****（logic-free markup），会增加应用程序复杂性，使其难于测试，代码无法重用。最近出现的一些DSL，如Mustache开始扭转这种局面，它们适用于多种语言和平台，并且允许使用任意工具编辑，无需额外的语言支持，对UI开发和应用程序的总体设计很有帮助。

## 应考虑使用Gradle替代Maven。

****Maven****一直是Java领域自动化构建的主打产品。然而，由于其缺少灵活性和自动化最佳实践的支持，尤其是在持续交付领域，应考虑使用Gradle替代Maven。

## ****领域特定语言****（Domain Specific Language, DSL）声

## ****声明性配置****（declarative provisioning）

## ****Highcharts**** 互动图表可视化效果

JavaScript 引擎的性能提升结合着广泛被采纳的基于 HTML 的嵌入式 SVG 文档，这使得基于纯 Javascript 的客户端图表和可视化解决方案被更多人所接受。****Highcharts**** 是我们尝试过的优秀者之一，它具备对高可配置性互动图表的灵活支持，并可轻而易举地对大数据集进行渲染。

****D3**** 是将数据集绑定至 DOM 的 JavaScript 函数库，通过声明式的方法将文档转换为不同的可视化效果（从图表到热成像图）。对 HTML、CSS 和 SVG 的支持，和采用可扩展的插件模型，我们喜欢它所提供的通过多种直观方法表现数据的方式。

## MongoDB

在设计领域模型时，聚合模式有助于结构和模块化。映射到关系型数据库，这种聚合在数据库表中时不可见的。文档型数据库，例如MongoDB，则可以将你的****聚合建模成文档****。这种1：1的映射意味着，聚合的根应该是从集合中加载的对象。

## ****JavaScript作为一个平台****(JavaScript as a platform)

# 不推荐

*：******WS-*****技术应谨慎使用