关于Java框架Vert.x的几点思考

发表于2015-05-20 17:18| 2652次阅读| 来源CSDN| 0 条评论| 作者白小勇

[特别策划](http://www.csdn.net/tag/%E7%89%B9%E5%88%AB%E7%AD%96%E5%88%92/news)[Java](http://www.csdn.net/tag/java/news)[Java20周年](http://www.csdn.net/tag/java20%E5%91%A8%E5%B9%B4/news)[《程序员》电子刊](http://www.csdn.net/tag/%E3%80%8A%E7%A8%8B%E5%BA%8F%E5%91%98%E3%80%8B%E7%94%B5%E5%AD%90%E5%88%8A/news)

**摘要：**Vert.x是一个基于JVM、轻量级、高性能的应用平台，非常适用于最新的移动端后台、互联网、企业应用架构。Vert.x基于全异步Java服务器Netty，并扩展出了很多有用的特性。

**Vert.x简介**

在Java20周年之际，Java用户对Java的抱怨与日俱增，比如内存管理、笨重的JavaEE等。而Java依然在TIOBE编程语言排行榜上艰难的维持第一名的位置，随着一些新编程语言的兴起，这个领域目前呈现一种混战的态势。

在这种背景下，Java届的小鲜肉框架——Vert.x于2015年5月7日发布了3.0-milestone5版本，距离计划6月22日发布Vert.x3.0.0-final越来越近了，Vert.x用户组的粉丝们近期已经迫不及待地在宇宙中心（注：北京五道口）组织了一次Vert.x中国用户组Meetup，针对Vert.x工程化开发问题以及Vert.x3新特性展开了探讨。Vert.x（http://vertx.io/）是一个基于JVM、轻量级、高性能的应用平台，非常适用于最新的移动端后台、互联网、企业应用架构。

Vert.x基于全异步Java服务器Netty，并扩展出了很多有用的特性。Vert.x的亮点有：

**同时支持多种编程语言—**—目前已经支持了Java、JavaScript、Ruby、Python、Groovy、Clojure、Ceylon等。对程序员来说，直接好处就是可以使用各种语言丰富的LIB，同时也不再为编程语言选型而纠结；

**异步无锁编程——**经典的多线程编程模型能满足很多Web开发场景，但随着移动互联网并发连接数的猛增，多线程并发控制模型性能难以扩展，同时要想控制好并发锁需要较高的技巧，目前Reactor异步编程模型开始跑马圈地，而Vert.x就是这种异步无锁编程的一个首选；

**对各种IO的丰富支持**——目前Vert.x的异步模型已支持TCP、UDP、FileSystem、DNS、EventBus、Sockjs等；

**极好的分布式开发支持**——Vert.x通过EventBus事件总线，可以轻松编写分布式解耦的程序，具有很好的扩展性；

**生态体系日趋成熟**——Vert.x归入Eclipse基金会门下，异步驱动已经支持了Postgres、MySQL、MongoDB、Redis等常用组件，并且有若干Vert.x在生产环境中的应用案例。

**Reactor模式**

和传统Java框架的多线程模型相比，Vert.x Netty是 Reactor模式的Java实现。考古了一下Reactor模式， 其理论最早由Washington University的Douglas C. Schmidt教授在1995年提出，在《Proactor - An Object Behavioral Pattern for Demultiplexing and Dispatching Handlers for Asynchronous Events 》这篇论文中做了 完整介绍。

图1-6是对其关键原理部分展开分析。

图1 一个经典Web Server在收到Web浏览器请求后的处理过程

图2 一个经典Web Server使用多线程模型，并发处理来自多个Web浏 览器的请求

图3 Web浏览器连接到一个Reactor模式的Web Server处理过程。利 用了Initiation Dispatcher组件，把耗时的IO操作事件注册到Initiation Dispatcher组件

图4 Web浏览器访问一个Reactor模式的Web Server处理过程。耗时IO 操作由其它线程执行，IO执行完成后通知Initiation Dispatcher，再回到 Http Handler执行

图5 Web浏览器连接一个Proactor模式的Web Server处理过程。和Reactor的主要区别是耗时IO操作交给操作系统异步IO库执行（例如 GNU/Linux aio），操作系统异步IO库执行完毕后，通过异步IO通知机制（例如epoll）触发Completion Dispatch，再交给Http Handler执行

图6 Web浏览器访问一个Proactor模式的Web Server处理过程。和Reactor的主要区别是耗时IO操作交给操作系统异步IO库执行（例如 GNU/Linux aio），操作系统异步IO库执行完毕后，通过异步IO通知机制（例如epoll）触发Completion Dispatch，再交给Http Handler执行

事实上，Vert.x/Netty的Reactor实现部分是在Netty 4.0如上述所示的代码中实现，和上述图中能对应的几个类是io.netty.channel.nio.NioEventLoop，io.netty. channel.epoll.EpollEventLoop，java.nio.channels.spi.SelectorProvide。

**Vert.x3.0的更新**  
  
Vert.x3.0是对Vert.x2.x的重大升级，不仅仅是package从org.vertx到io.vertx的全面替换，一些重要的核心类也都做了破坏式的重构，几乎很难从vert.x2程序升级到vert.x3.0程序。建议新项目直接从Vert.x3.0开始。以下是Vert.x3的一些功能升级：

* Vert.x2.x中的模块体系去掉了。目前Vert.x3.0推荐用Maven的模块体系，当然不仅限于Maven；支持其他语言在Vert.x上的代码生成；
* Vert.x3.0项目构建，从Gradle改为Maven；为了更好地利用Java8的Lambdas表达式，只支持Java8；默认采用扁平的classpath结构；
* Verticle工厂方式简化；支持用编程的方式实例化Verticle、以及部署Verticle实例；当你阻塞Eventloop主线程时警告，阻塞Reactor主线程是一种错误的使用方式；移除了PlatformManager模块；集群管理可以用编程的方式调用支持集群节点之间的共享数据；完全重写了HTTPclient，更完善；
* WebSocketAPI改善；
* SSL/TLS的改善；
* Eventbus的API改善；
* 支持Eventbus代理；增加了扩展项目集'ext'stack；
* 增加了MongoService，支持MongoDB的纯异步驱动；
* 实现ReactiveStreams；
* 对reactive-streams的实现；
* 支持Options类的使用，可以构造函数带参数进去；

更完整的样例工程。请见：<https://github.com/vert-x3/example-proj>

**综述**

综合来看，Vert.x3还未正式发布，但其主体功能已经开发完毕并趋于稳定，在应用开发中已经可以考虑使用。程序员在适应了异步回调式的编程方式后，相信很快可能感受到Reactor模式的性能红利和Vert.x的魅力。也许Vert.x会给Java和应用框架领域带来不一样的惊喜。

白小勇

**作者简介：**白小勇，炼石网络CipherGateway创始人、CEO，专注公有云安全。曾就职 于中国数码集团负责研发CMS、在用友工程负责研发ERP和代码 生成器平台、用友电信事业部研发平台总监，毕业于北京理工大 学。关注领域：密码应用、数据处理引擎技术、云平台、快速应 用开发平台。

PS：对Vert.x感兴趣的开发者朋友，可以移步去[Vert.x Meetup](http://www.meetup.com/Vertx-Beijing/events/222448854/)，现场与大牛们零距离探讨技术问题，聆听Vert.x的最佳实践，您也可以成为vert.x模块的贡献者。