响应式Web框架实战

大数据具有4V(Volume,Variety,Velocity,Value)特征，其中以RxJava和Reactor为代表的响应式编程技术专门针对高并发（Velocity）问题。在Wiki中，Reactive的定义如下：

*In computing, reactive programming is an asynchronous programming paradigm concerned with data streams and the propagation of change.*

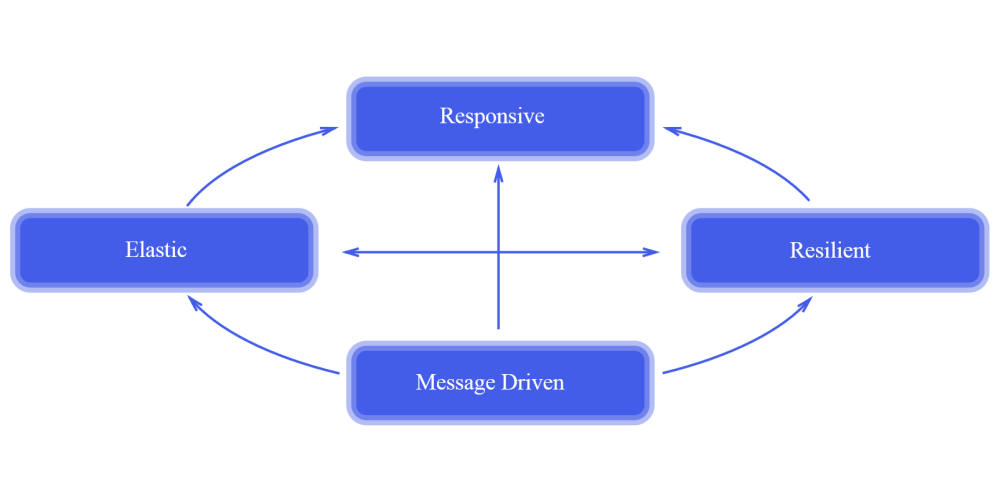
在上述响应式编程的定义中，除了异步编程，还包含两个重要的关键词：

* Data streams，即数据流，分为静态数据流（比如数组、文件）和动态数据流（比如事件流、日志流）两种。基于数据流模型，RP得以提供同一的Stream风格的处理接口，而且允许复用和同时接入多个订阅者
* The progagation of change，变化传播，简单来说就是以数据流为输入，经过一连串操作转化为另一个数据流，然后分发给各个订阅者的过程。有点类似于函数式编程中的组合函数，将多个函数串联起来，把一组输入数据转化为格式不同的输出数据。

从本质上说，PR是一种异步编程框架，其至少包含以下三个特性：

1. 描述而非执行，在最终调用subscribe方法之前，从发布端到订阅端是lazy计算模式，不会进行数据传输
2. 提高吞吐量，类似于HTTP/2的连接复用，RP通过线程复用来提高吞吐量，对于Spring 5中运行在Netty异步容器中的RP应用，用于请求的线程数是可配置
3. Backpressure支持，负载反馈机制，订阅者根据自己的处理能力向发布者请求响应数量的数据，但是不同于订阅者pull模式，订阅者只需要发送一次请求，在发送过程中发布者根据反馈调整发送速度。

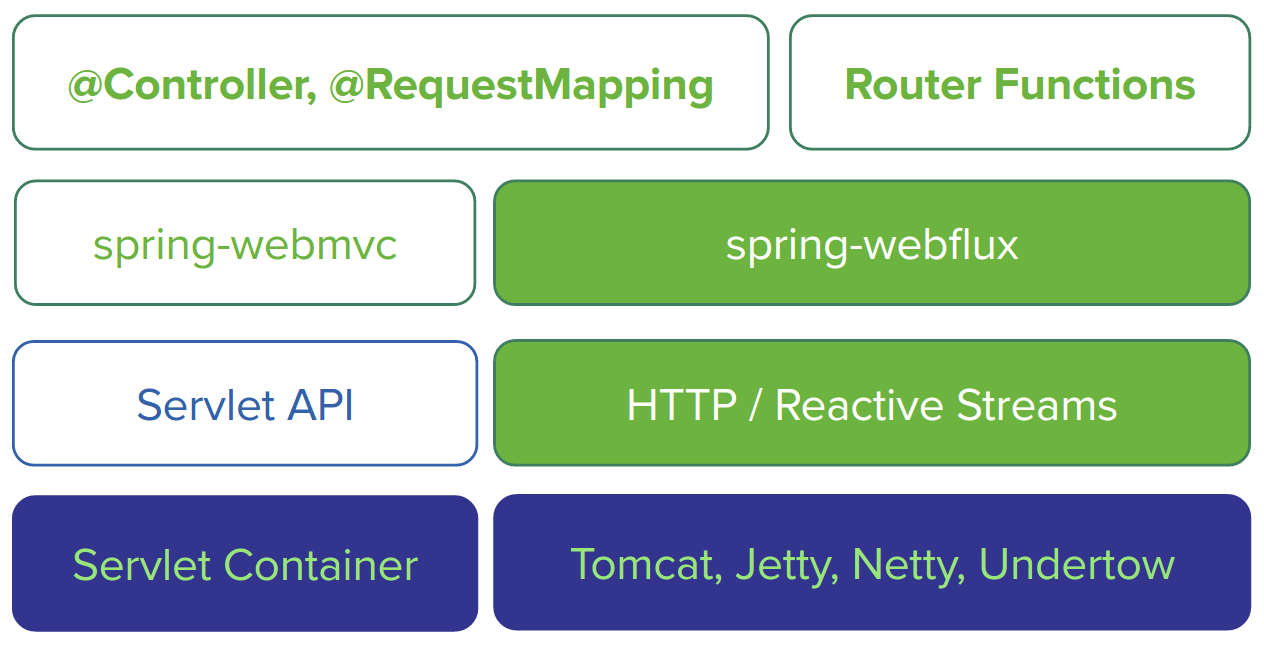
响应式解决问题如下：



1. Responsive，可响应的，要求系统尽可能的做到任何时间都能够及时响应
2. Resilient，可恢复的，要求系统即使出错，也能保持可响应性
3. Elastic，可伸缩性，要求系统在各种负载下都能保持可响应性
4. Message Driven，消息驱动，要求系统通过异步消息连接各个组件

对于任何一个响应式系统，首先要保证可响应性。

在Spring 5中提供了完整的端到端的响应式编程的支持：



左侧是传统的基于Servlet的Spring Web MVC框架，右侧是5.0版本新引入的基于Reactive Streams的Spring WebFlux框架，从上到下以此是Router Functions，WebFlux和ReactiveStreams 三个新组件：

* Router Functions，对标@Controller,@RequestMapping等标准Spring MVC注解，提供一套函数式风格的API，用于创建Router/Handler和Filter
* WebFlux，核心组件，协调上下游各个组件提供响应式编程支持
* Reactive Streams，支持背压的异步数据流处理标准，主流实现有RxJava和Reactor，Spring WebFlux默认集成的是Reactor

在Web容器的选择上，Spring WebFlux支持像Tomcat/Jetty这样传统容器，由支持Netty，Undertow异步容器，在所有容器中，Spring WebFlux都会将其输入输出流适配成Flux<Data Buffer>格式，以便进行统一处理。

https://github.com/LearningByExample/reactive-ms-example

https://github.com/ggonchar/reactive-spring5-example

<https://github.com/mutexkid/rxjava-webservice-example>

<https://github.com/emac/spring5-features-demo>

<https://github.com/funktionio/funktion>

https://github.com/LearningByExample/reactive-ms-example

https://blog.csdn.net/emac/article/details/75269929

http://emacoo.cn/backend/spring5-reactive-tutorial/

http://emacoo.cn/backend/spring5-overview/

http://emacoo.cn/backend/reactive-overview/