

High performance reactive applications with Vert.x

Tim Fox

Red Hat

Bio

- Employed By Red Hat to lead the Vert.x project
- Worked in open source exclusively for the past 9 years
- Some projects I've been involved with: Vert.x (creator), RabbitMQ, HornetQ (creator), JBoss AS, Mobicents...

Current Status

- Vert.x 2.1.2 current production release
- Vert.x 3.0 in development (release towards end of year)
- This talk will describe features in Vert.x 3.0 – not all available in Vert.x 2.x

What is Vert.x?

- General purpose JVM application framework
- Inspiration from Erlang/OTP and Node.js - Non blocking and asynchronous
- Polyglot – we don't force you to use one language.
- Very high performance
- Simple but not simplistic
- Lightweight
- Winner of JAX innovation award 2014 for most innovative Java technology!

Polyglot

Full implementation:



Core Asynchronous APIs

- Core is small and static
- TCP/SSL clients and servers
- HTTP/HTTPS clients and servers
- Websockets, SockJS
- File system
- Event bus
- DNS
- UDP
- Distributed Data

Why Asynchronous?

- Modern servers need to handle high levels of concurrency – web servers, websockets, IoT etc
- OS threads are still a precious resource
- Need to service many connections with small number of threads
- Blocked OS threads means they can't do other work

Verticle

- Execution unit of Vert.x
- Can be written in any language
- Single threaded – less scope for race conditions
- Verticles communicate by message passing
- Hmm.. sounds like the Actor Model?
- Can be run at command line or embedded in an app.
- Can be packaged into jars and like in Maven, bintray etc

Demo

Event Bus

- The nervous system of Vert.x
- Verticles send messages over the event bus
- Point to point. Publish/Subscribe. Request/Response
- Pass strings, buffers, primitive types or JSON
- Can provide codecs for custom types
- JSON messages are “convention” for structured data

Clustered Event Bus

- Lightweight peer-to-peer messaging system
- Connects multiple Vert.x JVM instances
- Applications are loosely coupled components distributed across your network
- No monolithic “application server”
- Micro-services

Event bus in the Browser

- Event bus extends to *client side* JavaScript too
- Uses the same API on the client
- Powerful distributed event space spanning both client and server nodes
- Ideal for modern “real-time” web applications
- Use whatever client side toolkit you prefer

Demo

Running verticles

- Verticles can be run directly on command line
- Can be trivially embedded
- Can be packaged into executable “fat” jars

```
public static void main(String[] args) {  
    Vertx vertx = Vertx.vertx();  
    vertx.deployVerticle("java:com.mycompany.MyVerticle");  
}
```

Demo

High Availability

- Automatic failover of deployed modules
- Nodes can be logically grouped
- Network partition detection (quorum)

Demo

Code Generation

- Manually maintaining N language APIs a lot of work!
- Vert.x 3.0 generates these APIs from the Java API
- Uses MVEL 2 templates
- Constraints on interfaces to make this feasible

Rx-ify that!

- Sick of callback hell?
- We will use codegen to generate Rx-ified versions of our APIs
- RxJava – Reactive Extensions for the JVM

Distributed Data

- Distributed Asynchronous Map
- Distributed Asynchronous Lock
- Distributed Asynchronous Counter

```
map.get("someKey", ar -> {});
```

```
sharedData.getLock("lockName", ar -> {});
```

```
counter.incrementAndGet(ar -> {});
```

Management and monitoring

- Metrics provided by many of the core objects
- Can be exposed via JMX or on the event bus

Official stack

- Databases – Postgres/MySQL, JDBC, Mongo, Redis, etc
- Messaging – JMS, AMQP
- IoT – MQTT, CoAP
- Authentication/Authorisation
- REST microservices
- etc

Summary

- Write apps as set of loosely coupled components that live *anywhere* where you want – no app server.
- Polyglot – use the language(s) you want
- Simple concurrency – wave goodbye to most race conditions
- Verticles – a library of lego bricks to build apps with
- High availability
- Ease of development

Project Info

- Independent Community Project
- The main project is an Eclipse Foundation project
- All code is on GitHub
- 100% open source (ASL 2.0 + Creative Commons)
- One of the most popular Java projects on GitHub

Get involved!

- Vert.x 3.0 in development now
- Active and growing community
- Find us on GitHub
- Google group: [vertx](#)
- IRC channel: [#vertx](#) on [freenode.net](#)

Q & A