

Vert.x

Zdeněk Merta, jOpenSpace 2013

Co je Vert.x?

- **aplikační platforma postavená nad JVM** (Java 7+)
- **vysoký výkon** - neblokující, udátně řízený, asynchronní model
- **škálovatelnost** - JVM přes dostupná jádra, cluster
- **vysoká dostupnost**
- **polyglot** - Java, JavaScript, Ruby, Python, Groovy, ...

Motivace

- **moderní real-time aplikace**
 - komunikační nástroje (sociální sítě, chat, ...)
 - kolaborační nástroje (Google Docs, ...)
 - online hry
- **C10K problém**
 - mnoho klientů 10 000+
 - mnoho dlouhotrvajících spojení

Reactor pattern

- aplikace je řízena událostmi
- na události se registrují handlers
- vlákno zpracovává události a spouští registrované handlers
- vlákno nesmí být blokováno!!!

Blokující operace

- **některé operace jsou přirozeně blokující**
 - volání tradičních API (např. JDBC)
 - dlouhotrvající operace (výpočty, generování, ...)

Hybridní model vláken

1. event loop

- Multi Reactor Pattern
- počet vláken standardně podle počtu procesorů
- vykonává neblokující operace

2. background worker pool

- pool vláken
- vykonává blokující nebo dlouhotrvající operace

Core API (100% asynchronní)

- **TCP/SSL server/klient**
- **HTTP/HTTPS server/klient**
- **Websockets server/klient, SockJS**
- **Event Bus / sdílená data**
- **časovače**
- **souborový systém**
- **konfigurace**
- **logování**

Verticle

- **nejmenší spustitelná jednotka** - třída/skript
- **vykonává neblokující operace**
- **spouští se v event loop** (vždy ve stejném)
- **Konkurence** - single-threaded
 - běží vždy pouze v jednom vlákně (odpadá synchronizace, zámky, ...)
 - izolace (vlastní classloader)
- **přístup ke Core API, registrace handlerů**
- **deploy dalších verticlů**

Worker verticle

- **vykonává blokuující operace**
- **spouští se ve vlákně z worker poolu**
- **vlákno může být pokaždé jiné**
- **vyjímečně může běžet ve více vláknech (multi-threaded)**

Moduly

- **znovupoužitelné komponenty** (mohou obsahovat více verticlů)
- **zip archiv obsahující kód, konfiguraci, závislosti**
- **veřejný repozitář modulů**

Instance Vert.x

- **běží ve vlastním JVM**
- **umožňuje spouštět verticly**
- **může běžet v clusteru**
- **event loops, background worker pool**

Komunikace mezi verticly

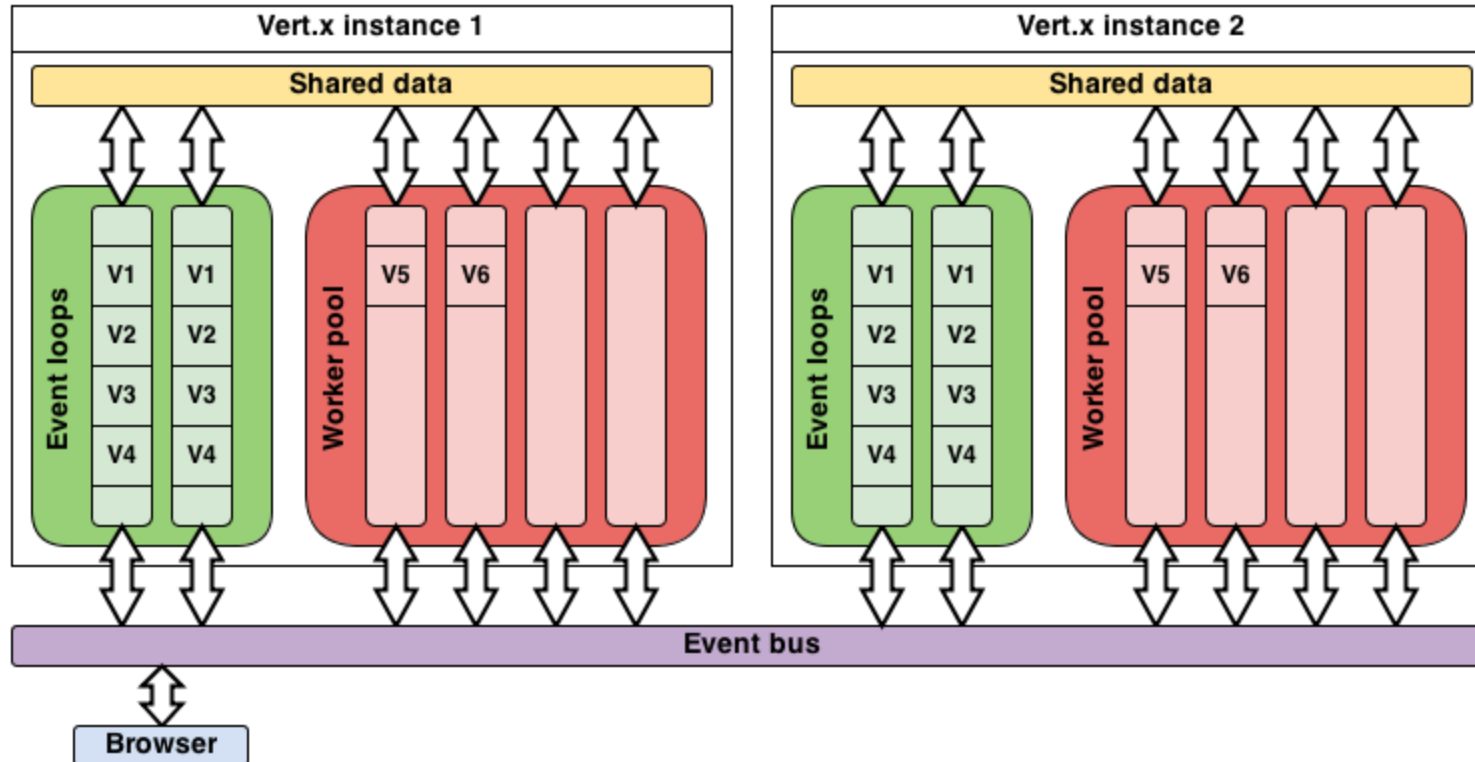
1. Event bus

- zprávy
- jednoduché adresování
- publish/subscribe, P2P, request/response
- distribuovaná, zasahuje i do browseru

2. sdílená data

- immutable objekty ukládané do sdílené mapy nebo množiny
- pouze v rámci jedné instance (zatím?)

Architektura



Nástroje

- **šablona pro Maven a Gradle**
- **podpora testů** (JUnit)
 - jednotkové
 - integrační
- **neexistuje podpora pro IDE**

Nevýhody

- **poměrně mladá platforma** (necelé dva roky vývoje)
- **menší komunita**
- **malý počet veřejných modulů**
- **nepřehledný kód v Javě** (vnitřní třídy)

Otázky?

Díky za pozornost