SERVICES RELOADED

INCREASED THROUGHPUT WITH PROJECT LOOM VIRTUAL THREADS

Kito Mann (@kito99) . virtua.tech
KITO D. MANN (@KITO99)

- Principal Consultant at Virtua (http://virtua.tech)
- Training, consulting, architecture, mentoring
  - Java microservices, Jakarta EE, Web Components, Angular
- Official US PrimeTek partner
  - PrimeNG, PrimeVue, PrimeFaces, PrimeReact
- Author, JavaServer Faces in Action
YOU
Tell me about yourself...
LET’S GO ON JAVA THREAD JOURNEY...
GREEN THREADS: JDK 1.1 - 1.2
(1997-2000)

- Each thread had several green threads
GREEN THREADS: JDK 1.1 - 1.2
(1997-2000)

- Each thread had several green threads
- M:1 scheduling
GREEN THREADS: JDK 1.1 - 1.2
(1997-2000)

- Each thread had several green threads
- M:1 scheduling
- https://en.wikipedia.org/wiki/Green_thread
JDK 1.3 - 20+ (2000-CURRENT)
JDK 1.3 - 20+ (2000-CURRENT)

- Platform threads
  - Each Java thread maps to an OS thread
JDK 1.3 - 20+ (2000-CURRENT)

- Platform threads
  - Each Java thread maps to an OS thread
- 1:1 scheduling
EVERYBODY BLOCK
EVERYBODY BLOCK

- Servlet API
  - Tomcat
  - Jetty
EVERYBODY BLOCK

- Servlet API
  - Tomcat
  - Jetty
- Apache Web Server
EVERYBODY ASYNC
EVERYBODY ASYNC

- NGINX (2004)
EVERYBODY ASYNC

- NGINX (2004)
EVERYBODY ASYNC

- NGINX (2004)
- Node.js (2009)
EVERYBODY ASYNC

- NGINX (2004)
- Node.js (2009)
- Akka (2010)
EVERYBODY ASYNC

- NGINX (2004)
- Node.js (2009)
- Akka (2010)
- Vert.x (2011)
  - Built on Netty but higher level
EVERYBODY ASYNC
EVERYBODY ASYNC

- Servlet 3.1 (Java EE 7, 2013) added async support
  - Not in JAX-RS / Jakarta RESTful services, though
EVERYBODY ASYNC
REACTIVE MANIFESTO

https://www.reactivemanifesto.org/
REACTING WITH APIS
REACTING WITH APIs

- ReactiveX (formerly Microsoft Reactive Extensions)
  - https://reactivex.io
  - .NET (2011)
  - RxJs
  - RxJava
  - others
REACTING WITH APIS
REACTING WITH APIS

- Spring Project Reactor (2013)
REACTING WITH APIS

- Spring Project Reactor (2013)
- Reactive Streams (2015)
REACTING WITH APIS

- Spring Project Reactor (2013)
- Reactive Streams (2015)
- Java 9 Reactive Streams (Flow interface) (2017)
REACTING WITH APIS

- Spring Project Reactor (2013)
- Reactive Streams (2015)
- Java 9 Reactive Streams (Flow interface) (2017)
- SmallRye Mutiny (2021)
CORE REACTIVE CONCEPTS
CORE REACTIVE CONCEPTS

- Observable
CORE REACTIVE CONCEPTS

- Observable
- Operators
CORE REACTIVE CONCEPTS

- Observable
- Operators
- Single
CORE REACTIVE CONCEPTS

- Observable
- Operators
- Single
- Subject
CORE REACTIVE CONCEPTS

- Observable
- Operators
- Single
- Subject
- Scheduler
REACTIVE SERVICES
REACTIVE SERVICES

- Quarkus
REACTIVE SERVICES

- Quarkus
- Spring Boot w/WebFlux
REACTIVE SERVICES

- Quarkus
- Spring Boot w/WebFlux
- Vert.x
REACTIVE SERVICES

- Quarkus
- Spring Boot w/WebFlux
- Vert.x
- Micronaut
REACTIVE SERVICES
package hello;

import org.springframework.data.redis.core.ReactiveRedisOperations;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;
import reactor.core.publisher.Flux;

@RestController
public class CoffeeController {
    private final ReactiveRedisOperations<String, Coffee> coffeeOps;

    CoffeeController(ReactiveRedisOperations<String, Coffee> coffeeOps) {
        this.coffeeOps = coffeeOps;
    }
}
ENTER REACTIVE

Mo Reactive, Mo Problems.
ENTER REACTIVE
Mo Reactive, Mo Problems.
ENTER REACTIVE
ENTER REACTIVE

Code is hard to read
ENTER REACTIVE
ENTER REACTIVE

Stack traces are hard to read
JDK 21+ (2023+)

Project Loom Virtual Threads
FRUIT OF THE LOOM
Project Loom: Fibers, Continuations, and Tail-Calls for the JVM

- [https://wiki.openjdk.org/display/loom/Main](https://wiki.openjdk.org/display/loom/Main)
- Latest JEP: [https://openjdk.org/jeps/444](https://openjdk.org/jeps/444)
FRUIT OF THE LOOM

- Project Loom: Fibers, Continuations, and Tail-Calls for the JVM
  - https://wiki.openjdk.org/display/loom/Main
  - Latest JEP: https://openjdk.org/jeps/444
- Started in 2017
FRUIT OF THE LOOM

- Project Loom: Fibers, Continuations, and Tail-Calls for the JVM
  - https://wiki.openjdk.org/display/loom/Main
  - Latest JEP: https://openjdk.org/jeps/444
- Started in 2017
- First Preview shipped with JDK 19
FRUIT OF THE LOOM

- Project Loom: Fibers, Continuations, and Tail-Calls for the JVM
  - https://wiki.openjdk.org/display/loom/Main
  - Latest JEP: https://openjdk.org/jeps/444
- Started in 2017
- First Preview shipped with JDK 19
- Final version slated for JDK 21 (Fall 2023)
FRUIT OF THE LOOM

• Project Loom: Fibers, Continuations, and Tail-Calls for the JVM
  ▪ https://wiki.openjdk.org/display/loom/Main
  ▪ Latest JEP: https://openjdk.org/jeps/444

• Started in 2017

• First Preview shipped with JDK 19

• Final version slated for JDK 21 (Fall 2023)

• M:N scheduling
FRUIT OF THE LOOM
FRUIT OF THE LOOM
HOW TO AVOID SPOILED FRUIT
HOW TO AVOID SPOILED FRUIT

• Don’t pool virtual threads
HOW TO AVOID SPOILED FRUIT

- Don’t pool virtual threads
- Be careful with thread-local variables
HOW TO AVOID SPOILED FRUIT

• Don’t pool virtual threads
• Be careful with thread-local variables
• Be aware of when code pins a virtual thread to its carrier thread
  ▪ static blocks
  ○ If used frequently, consider using java.util.concurrent.locks.ReentrantLock instead
HOW TO AVOID SPOILED FRUIT

- Don’t pool virtual threads
- Be careful with thread-local variables
- Be aware of when code pins a virtual thread to its carrier thread
  - static blocks
    - If used frequently, consider using java.util.concurrent.locks.ReentrantLock instead
- Use of JNI or foreign functions
VIRTUAL THREAD APIS

try (var executor = Executors.newVirtualThreadPerTaskExecutor(
    IntStream.range(0, 10_000).forEach(i -> {
        executor.submit(() -> {
            Thread.sleep(Duration.ofSeconds(1));
            return i;
        });
    }));
} // executor.close() is called implicitly, and waits
VIRTUAL THREAD APIS

Thread thread = Thread.ofVirtual().name("duke").unstarted(runnable);
Thread.startVirtualThread(Runnable);
Thread.isVirtual();
FRUIT OF THE LOOM
GROUNDWORK
FRUIT OF THE LOOM GROUNDWORK

- JVM internals reworked to support virtual threads
  - Thread API
  - java.util.concurrent
  - java.net and java.nio.channels
  - java.io
FRUIT OF THE LOOM
GROUNDWORK
FRUIT OF THE LOOM
GROUNDWORK
FRUIT OF THE LOOM
GROUNDWORK
FRUIT OF THE LOOM
GROUNDWORK
FRUIT OF THE LOOM
GROUNDWORK

- JEP 353 (Reimplement the Legacy Socket API) in JDK 13, and JEP 373 (Reimplement the Legacy DatagramSocket API) in JDK 15, replaced the implementations of java.net.Socket, ServerSocket, and DatagramSocket with new implementations designed for use with virtual threads.
FRUIT OF THE LOOM
GROUNDWORK
FRUIT OF THE LOOM GROUNDWORK

- JEP 418 (Internet-Address Resolution SPI) in JDK 18 defined a service-provider interface for host name and address lookup. This will allow third-party libraries to implement alternative java.net.InetAddress resolvers that do not pin threads during host lookup.
SERVLETS / WEB SERVER SUPPORT
SERVLET / WEB SERVER SUPPORT

- Tomcat 10.1.1 (2023)

  - Configure the StandardVirtualThreadExecutor
    https://tomcat.apache.org/tomcat-10.1-doc/config/executor.html#Virtual_Thread_Implemen
SERVLET / WEB SERVER SUPPORT

- Tomcat 10.1.1 (2023)
  - Configure the StandardVirtualThreadExecutor
    https://tomcat.apache.org/tomcat-10.1-doc/config/executor.html#Virtual_Thread_Implementation

- Jetty 10.0.12 and 11.0.12 and higher
  - https://webtide.com/jetty-12-virtual-threads-support
SERVLET / WEB SERVER SUPPORT
SERVLET / WEB SERVER SUPPORT

- Netty
  - Not currently; see: https://github.com/netty/netty/issues/12848
SERVLET / WEB SERVER SUPPORT

- Netty
  - Not currently; see: https://github.com/netty/netty/issues/12848
- Undertow ❌
MICROSERVICE FRAMEWORKS
MICROSERVICE FRAMEWORKS

- Quarkus (since 2022)
  - https://quarkus.io/guides/virtual-threads
  - Performance almost as good as reactive
package org.acme.rest;

import org.acme.fortune.model.Fortune;
import org.acme.fortune.repository.FortuneRepository;
import io.smallrye.common.annotation.RunOnVirtualThread;
import io.smallrye.mutiny.Uni;

import jakarta.ws.rs.GET;
import jakarta.ws.rs.Path;
import java.util.List;
import java.util.Random;

@Path("/")
public class FortuneResource {
MICROSERVICE FRAMEWORKS
MICROSERVICE FRAMEWORKS

- Helidon 4 (end of 2023)
  - Nima replacing Netty: https://helidon.io/nima
MICROSERVICE FRAMEWORKS

- Helidon 4 (end of 2023)
  - Nima replacing Netty: https://helidon.io/nima
- Micronaut 4.0.0 (2023)
  - https://micronaut.io/2023/07/14/micronaut-framework-4-0-0-0-released/
MICROSERVICE FRAMEWORKS

- Spring Boot
  
  - Just use the latest Tomcat and turn it on

```java
@Bean(TaskExecutionAutoConfiguration.APPLICATION_TASK_EXECUTOR
public AsyncTaskExecutor asyncTaskExecutor() {
    return new TaskExecutorAdapter(Executors.newVirtualThreadPerT
}

@Bean
public TomcatProtocolHandlerCustomizer<?> protocolHandlerVirtu
return protocolHandler -> {
    protocolHandler.setExecutor(Executors.newVirtualThreadPerT
};
```
JAKARTA EE
JAKARTA EE

- Currently being planned for Jakarta EE 11
WHAT ABOUT JDBC DRIVERS?
WHAT ABOUT JDBC DRIVERS?

- Oracle’s JDBC driver has been updated
  - [https://medium.com/oracledevs/introduction-to-oracle-jdbc-21c-driver-support-for-virtual-threads-189b918c56f4](https://medium.com/oracledevs/introduction-to-oracle-jdbc-21c-driver-support-for-virtual-threads-189b918c56f4)
WHAT ABOUT JDBC DRIVERS?

- Oracle’s JDBC driver has been updated
  - [https://medium.com/oracledevs/introduction-to-oracle-jdbc-21c-driver-support-for-virtual-threads-189b918c56f4](https://medium.com/oracledevs/introduction-to-oracle-jdbc-21c-driver-support-for-virtual-threads-189b918c56f4)

- Others?