Robert Jacko

Java Virtual Threads

CIKLUM

Project Loom

CIKLUM

easy-to-use, high-throughput lightweight concurrency (first commit 2007)

JEP 444: Virtual Threads
https://openjdk.org/jeps/444

JEP 428: Structured Concurrency
https://openjdk.org/jeps/428

Thread

construct of operating system (1967) to achieve concurrency

large stack and other resources that are maintained by the operating system – expensive context switching

process vs thread

Threads in Java

java.lang.Thread
since 1.0

mapped 1:1 to kernel threads scheduled by the operating system

Runnable 1.0 vs Callable 1.5

```
new Thread(() -> System.out.println("I am alive")).start();
```

Virtual Threads

Java 19 preview, Java 20 second preview, Java 21!

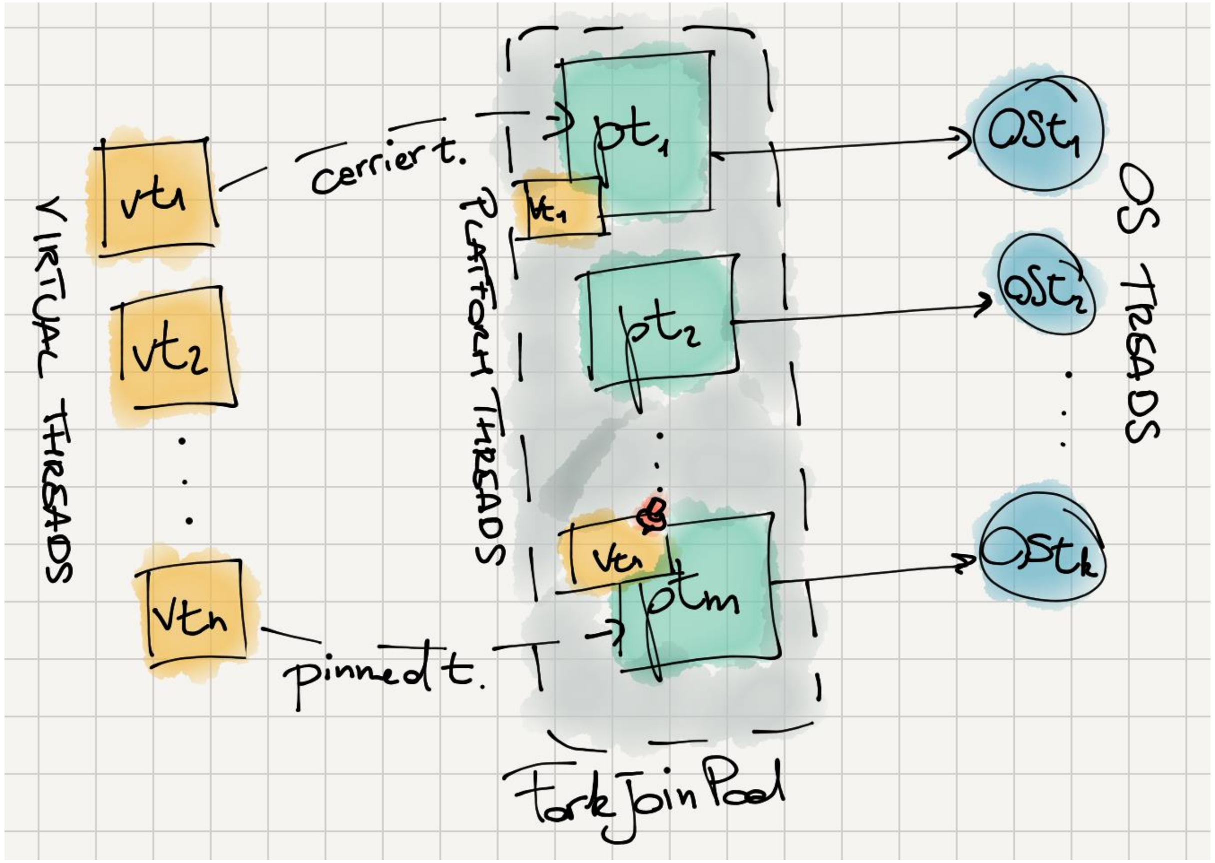
JVM construct

platform thread vs virtual thread

java.lang.VirtualThread
Thread -> BaseVirtualThread -> VirtualThread

default scheduler - ForkJoinPool

Virtual Threads



Virtual Threads

```
Thread.startVirtualThread(runnable);

Thread.ofVirtual().name(name).start(runnable);

try (var executor =
Executors.newVirtualThreadPerTaskExecutor()) {
    executor.submit(() -> {
        System.out.println("I am virtual and alive.");
    });
}
```

Virtual Threads

```
final ThreadFactory factory = Thread.ofVirtual().factory();
try (var executor =
Executors.newThreadPerTaskExecutor(factory)) {
 executor.submit(() -> {
        System.out.println("I am virtual and alive.");
    });
}
```

Advantages

JVM schedules execution not the system

less resources, faster context switching

(almost) no blocking operations

virtual threads vs reactive programming

increased throughput

Limitations

concurrency and performance based on CPUs/cores

synchronized blocks

- replace by ReentrantLock, ReadWriteReentrantLock, StampedLock

native method or a foreign function (JNI)

ThreadLocal

platform threads are the default

Let’s see some code

CIKLUM

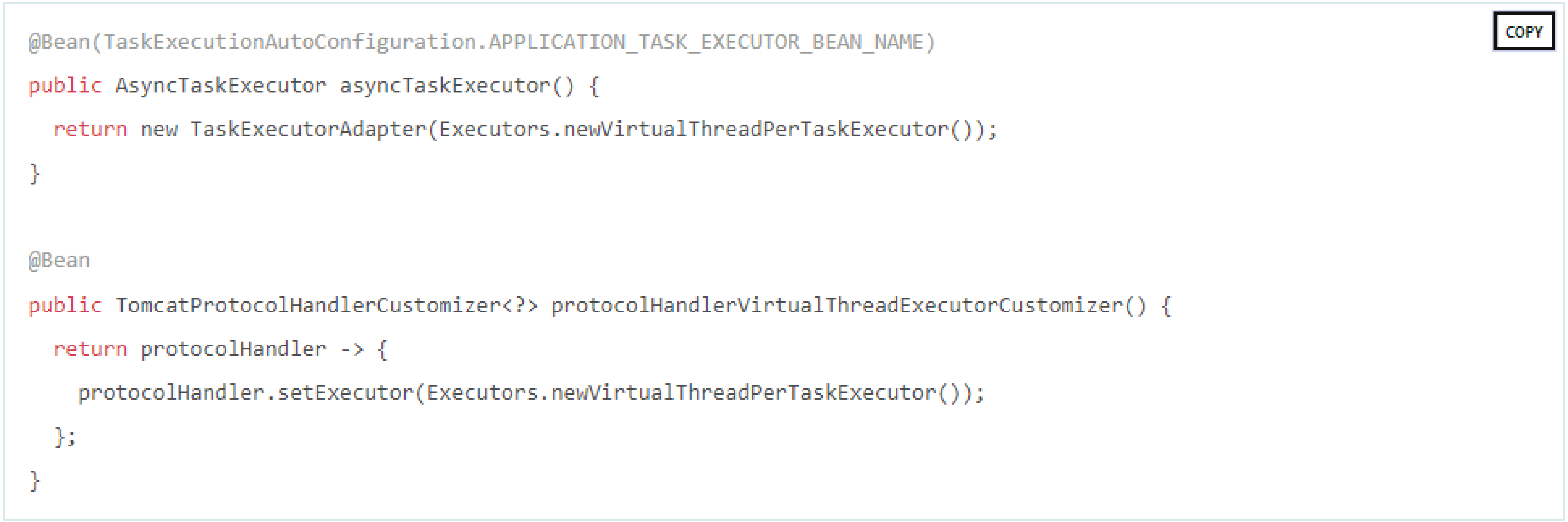
https://github.com/robo-jacko/virtual_threads

Virtual Threads Support

Helidon Níma
Spring Boot
Tomcat
Quarkus
...

Virtual Threads Support

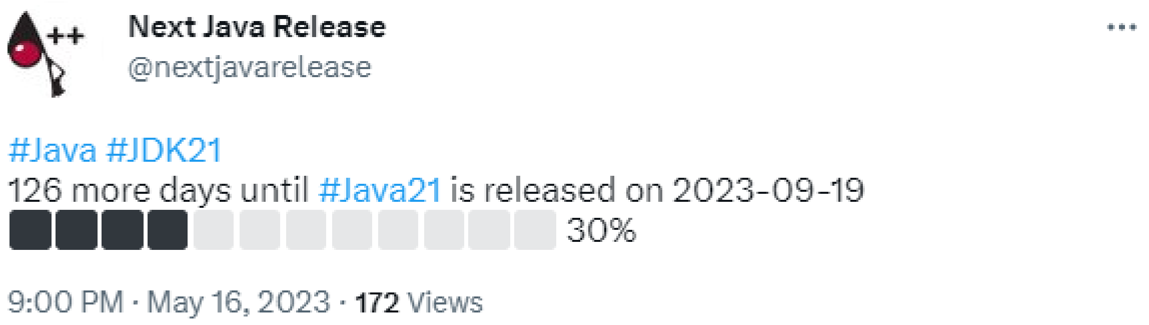
Spring Boot



```java
@Bean(TaskExecutionAutoConfiguration.APPLICATION_TASK_EXECUTOR_BEAN_NAME)
public AsyncTaskExecutor asyncTaskExecutor() {
  return new TaskExecutorAdapter(Executors.newVirtualThreadPerTaskExecutor());
}

@Bean
public TomcatProtocolHandlerCustomizer<?> protocolHandlerVirtualThreadExecutorCustomizer() {
  return protocolHandler -> {
    protocolHandler.setExecutor(Executors.newVirtualThreadPerTaskExecutor());
  };
}
```

Java 21


Next Java Release
@nextjavarelease

#Java #JDK21
126 more days until #Java21 is released on 2023-09-19
■■■■□□□□□□□□ 30%

9:00 PM · May 16, 2023 · 172 Views

Questions?

References

JEP 444: Virtual Threads
https://openjdk.org/jeps/444

JEP 453: Structured Concurrency
https://openjdk.org/jeps/453

The Ultimate Guide to Java Virtual Threads
https://blog.rockthejvm.com/ultimate-guide-to-java-virtual-threads/

References

Kotlin Coroutines vs Java Virtual Threads
https://itnext.io/kotlin-coroutines-vs-java-virtual-threads-a-good-story-but-just-that-91038c7d21eb

WRITING SIMPLER REACTIVE REST SERVICES WITH QUARKUS VIRTUAL THREAD SUPPORT
https://quarkus.io/guides/virtual-threads

https://github.com/robo-jacko/virtual_threads