The Value Of Reactive

Stephane Maldini - Reactive Engineering @ Pivotal
@smaldini

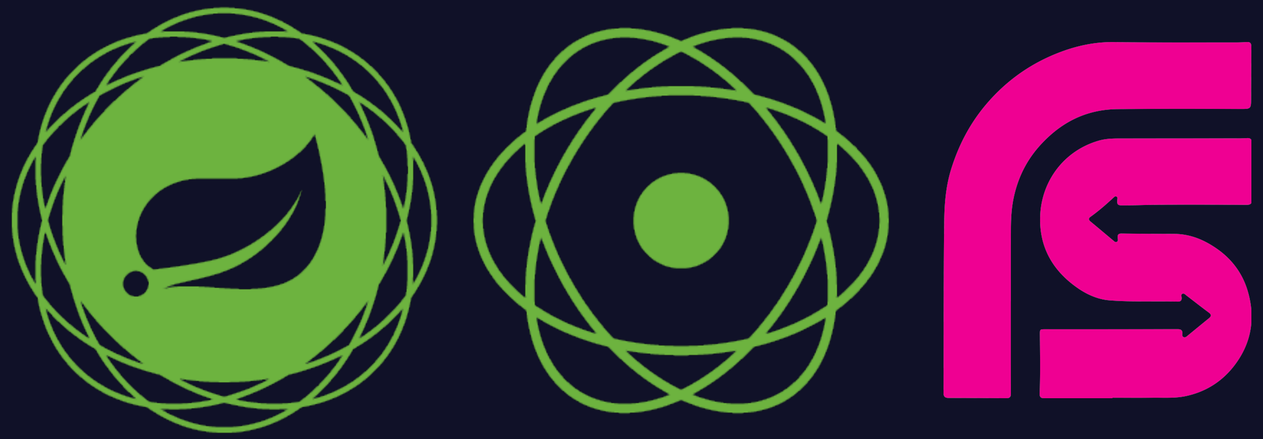
The Value Of Reactive

— What is it ? Where is the value ?

+

=

Pivotal®



R2DBC

No, I do not have a medium blog

Reactive Programming

- The next frontier for high-efficiency applications
- Non-blocking and allows for concurrent executions
- Often associated with functional programming
- Have no opinion on async and many flows are totally synchronous
- Resilient with graceful error handling and interruptions (producer or consumer related)

Reactive Programming

- The next frontier for high-efficiency applications
- Non-blocking and allows for concurrent executions
- Often associated with functional programming
- Have no opinion on async and many flows are totally synchronous
- Resilient with graceful error handling and interruptions (producer or consumer related)

A reactive API primer

```
@GetMapping("/health")
public Mono<Health> compositeHealth() {
  return Mono.zip(
      webClient.get()
               .uri(“https://alpha-service/health")
               .retrieve()
               .bodyToMono(Health.class),
      webClient.get()
               .uri("https://bravo-service/health")
               .retrieve()
               .bodyToMono(Health.class),
      (alpha, bravo) -> composite(alpha, bravo))
             .switchIfEmpty(emptyComposite());
}
```

But, is it only a writing style issue ?

Debugging Reactive flows

```
java.lang.RuntimeException: Am I looking like a nice stacktrace friends ?
    at io.spring.workshop.tradingservice.TradingCompanyClient.lambda$getTradingCompany$0(TradingCompanyClient.java:38) ~[classes/:na]
    at reactor.core.publisher.FluxPeekFuseable$PeekFuseableSubscriber.onNext(FluxPeekFuseable.java:190) ~[reactor-core-3.3.0]
    at reactor.core.publisher.Operators$MonoSubscriber.complete(Operators.java:1575) ~[reactor-core-3.3.0]
    at reactor.core.publisher.MonoFlatMap$FlatMapInner.onNext(MonoFlatMap.java:241) ~[reactor-core-3.3.0]
    at reactor.core.publisher.FluxOnAssembly$OnAssemblySubscriber.onNext(FluxOnAssembly.java:349) ~[reactor-core-3.3.0]
    at reactor.core.publisher.FluxMapFuseable$MapFuseableSubscriber.onNext(FluxMapFuseable.java:121) ~[reactor-core-3.3.0]
    at reactor.core.publisher.FluxContextStart$ContextStartSubscriber.onNext(FluxContextStart.java:103) ~[reactor-core-3.3.0]
    at reactor.core.publisher.FluxMapFuseable$MapFuseableConditionalSubscriber.onNext(FluxMapFuseable.java:287) ~[reactor-core-3.3.0]
    at reactor.core.publisher.FluxFilterFuseable$FilterFuseableConditionalSubscriber.onNext(FluxFilterFuseable.java:331) ~[reactor-core-3.3.0]
    at reactor.core.publisher.Operators$MonoSubscriber.complete(Operators.java:1575) ~[reactor-core-3.3.0]
    at reactor.core.publisher.MonoCollectList$MonoCollectListSubscriber.onComplete(MonoCollectList.java:121) ~[reactor-core-3.3.0]
    at reactor.core.publisher.FluxMap$MapSubscriber.onComplete(FluxMap.java:136) ~[reactor-core-3.3.0]
    at reactor.core.publisher.FluxPeek$PeekSubscriber.onComplete(FluxPeek.java:252) ~[reactor-core-3.3.0]
    at reactor.core.publisher.FluxPeek$PeekSubscriber.onComplete(FluxPeek.java:252) ~[reactor-core-3.3.0]
    at reactor.core.publisher.FluxMap$MapSubscriber.onComplete(FluxMap.java:136) ~[reactor-core-3.3.0]
    at reactor.netty.channel.FluxReceive.terminateReceiver(FluxReceive.java:391) ~[reactor-netty-0.9.0]
    at reactor.netty.channel.FluxReceive.drainReceiver(FluxReceive.java:197) ~[reactor-netty-0.9.0]
    at reactor.netty.channel.FluxReceive.onInboundComplete(FluxReceive.java:338) ~[reactor-netty-0.9.0]
    at reactor.netty.channel.ChannelOperations.onInboundComplete(ChannelOperations.java:350) ~[reactor-netty-0.9.0]
    at reactor.netty.channel.ChannelOperations.terminate(ChannelOperations.java:399) ~[reactor-netty-0.9.0]
    at reactor.netty.http.client.HttpClientOperations.onInboundNext(HttpClientOperations.java:555) ~[reactor-netty-0.9.0]
...
```

Debugging Reactive flows

```
java.lang.RuntimeException: Am I looking like a nice stacktrace friends ?
    at io.spring.workshop.tradingservice.TradingCompanyClient.lambda$getT                                    8) ~[classes/:na]
    at reactor.core.publisher.FluxPeekFuseable$PeekFuseableSubscriber.onNe                                   0]
    at reactor.core.publisher.Operators$MonoSubscriber.complete(Operators.
    at reactor.core.publisher.MonoFlatMap$FlatMapInner.onNext(MonoFlatM
    at reactor.core.publisher.FluxOnAssembly$OnAssemblySubscriber.onNext
    at reactor.core.publisher.FluxMapFuseable$MapFuseableSubscriber.onNex                                    0]
    at reactor.core.publisher.FluxContextStart$ContextStartSubscriber.onNext(
    at reactor.core.publisher.FluxMapFuseable$MapFuseableConditionalSubsc                                    r-core-3.3.0]
    at reactor.core.publisher.FluxFilterFuseable$FilterFuseableConditionalSub                                ctor-core-3.3.0]
    at reactor.core.publisher.Operators$MonoSubscriber.complete(Operators.
    at reactor.core.publisher.MonoCollectList$MonoCollectListSubscriber.onC                                  e-3.3.0]
    at reactor.core.publisher.FluxMap$MapSubscriber.onComplete(FluxMap.
    at reactor.core.publisher.FluxPeek$PeekSubscriber.onComplete(FluxPeek
    at reactor.core.publisher.FluxPeek$PeekSubscriber.onComplete(FluxPeek
    at reactor.core.publisher.FluxMap$MapSubscriber.onComplete(FluxMap.
    at reactor.netty.channel.FluxReceive.terminateReceiver(FluxReceive.java
    at reactor.netty.channel.FluxReceive.drainReceiver(FluxReceive.java:197
    at reactor.netty.channel.FluxReceive.onInboundComplete(FluxReceive.jav
    at reactor.netty.channel.ChannelOperations.onInboundComplete(ChannelO
    at reactor.netty.channel.ChannelOperations.terminate(ChannelOperations
    at reactor.netty.http.client.HttpClientOperations.onInboundNext(HttpClien
...
```

Debugging Reactive flows

```
java.lang.RuntimeException: Am I looking like a nice stacktrace friends ?
    at io.spring.workshop.tradingservice.TradingCompanyClient.lambda$getT
    at reactor.core.publisher.FluxPeekFuseable$PeekFuseableSubscriber.onNe                    0]
    at reactor.core.publisher.Operators$MonoSubscriber.complete(Operators.
    at reactor.core.publisher.MonoFlatMap$FlatMapInner.onNext(MonoFlatM
    at reactor.core.publisher.FluxOnAssembly$OnAssemblySubscriber.onNext
    at reactor.core.publisher.FluxMapFuseable$MapFuseableSubscriber.onNex                     0]
    at reactor.core.publisher.FluxContextStart$ContextStartSubscriber.onNext(
    at reactor.core.publisher.FluxMapFuseable$MapFuseableConditionalSubsc                     or-core-3.3.0]
    at reactor.core.publisher.FluxFilterFuseable$FilterFuseableConditionalSub                 ctor-core-3.3.0]
    at reactor.core.publisher.Operators$MonoSubscriber.complete(Operators.
    at reactor.core.publisher.MonoCollectList$MonoCollectListSubscriber.onC                   e-3.3.0]
    at reactor.core.publisher.FluxMap$MapSubscriber.onComplete(FluxMap.
    at reactor.core.publisher.FluxPeek$PeekSubscriber.onComplete(FluxPeek
    at reactor.core.publisher.FluxPeek$PeekSubscriber.onComplete(FluxPeek
    at reactor.core.publisher.FluxMap$MapSubscriber.onComplete(FluxMap.
    at reactor.netty.channel.FluxReceive.terminateReceiver(FluxReceive.java
    at reactor.netty.channel.FluxReceive.drainReceiver(FluxReceive.java:197
    at reactor.netty.channel.FluxReceive.onInboundComplete(FluxReceive.jav
    at reactor.netty.channel.ChannelOperations.onInboundComplete(ChannelO
    at reactor.netty.channel.ChannelOperations.terminate(ChannelOperations
    at reactor.netty.http.client.HttpClientOperations.onInboundNext(HttpClien
...
```

You ?

Using blocking code in reactive callbacks

```
TradingCompany tradingCompany = this.restTemplate.exchange(
    get("http://localhost:8082/details/{ticker}", ticker),
    TradingCompany.class)
    .getBody();
```

Using blocking code in reactive callbacks

```
someMono.doOnSuccess(data -> {

     TradingCompany tradingCompany = this.restTemplate.exchange(
         get("http://localhost:8082/details/{ticker}", ticker),
         TradingCompany.class)
         .getBody();

     return tradingCompany;

});
```

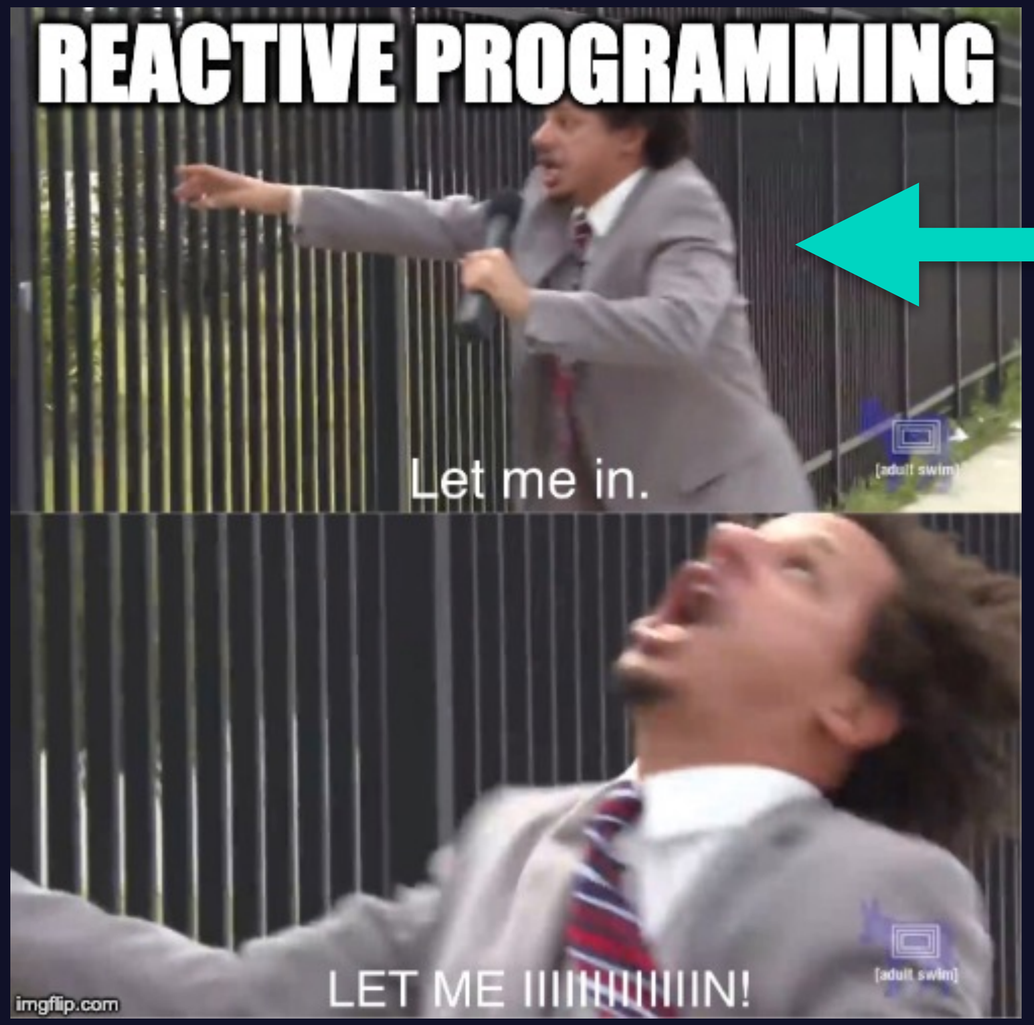
Using blocking code in reactive callbacks

```
someMono.doOnSuccess(data -> {

    TradingCompany tradingCompany = this.restTemplate.exchange(
        get("http://localhost:8080/details/{ticker}", ticker),
        TradingCompany.class)
        .getBody();

    return tradingCompany;

});
```


REACTIVE PROGRAMMING
Let me in.
[adult swim]
LET ME IIIIIIIIIIIIN!
[adult swim]
imgflip.com

You

What are you priorities ?

What are you priorities ?

Time To Market / Dev Productivity

What are you priorities ?

Time To Market / Dev Productivity

Day 2 Operations / Observability

What are you priorities ?

Time To Market / Dev Productivity

Day 2 Operations / Observability

[...]

What are you priorities ?

Time To Market / Dev Productivity

Day 2 Operations / Observability

[...]

Running costs / efficiency

What are you priorities ?

Time To Market / Dev Productivity

Day 2 Operations / Observability

[...]

Running costs / efficiency  Reactive ?

Give me

Reasons

To use this stuff

Order(s) of Magnitude more **efficient**

Order(s) of Magnitude more **efficient**

- **Lower** Number of processes/threads used by instance
- **Lower** Memory Pressure and associated costs (Garbage collectors)
- **Lower** Hardware specification requirements
- **Lower** Startup time

Order(s) of Magnitude more **efficient**

Show me some numbers

Order(s) of Magnitude more **efficient**

Spring WebFlux.Fn - Reactor Netty - 4 threads

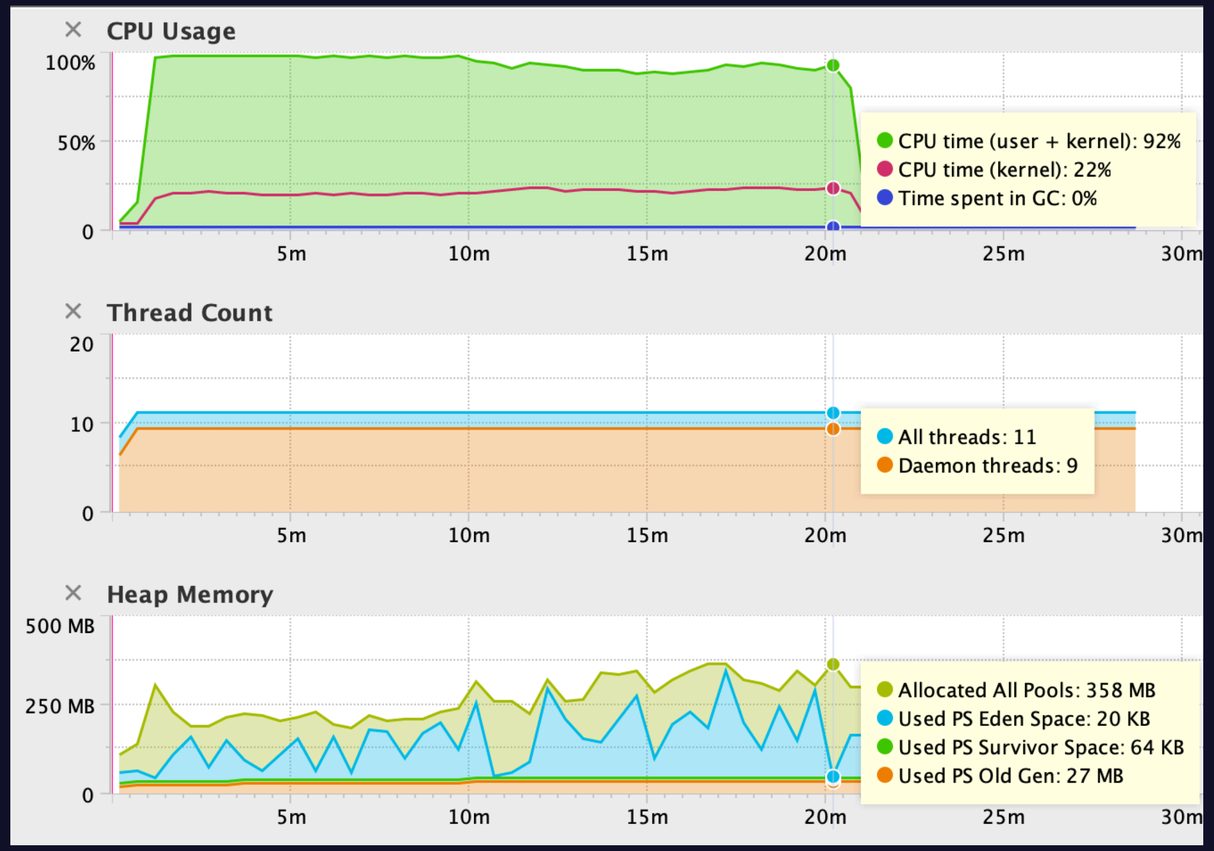
&

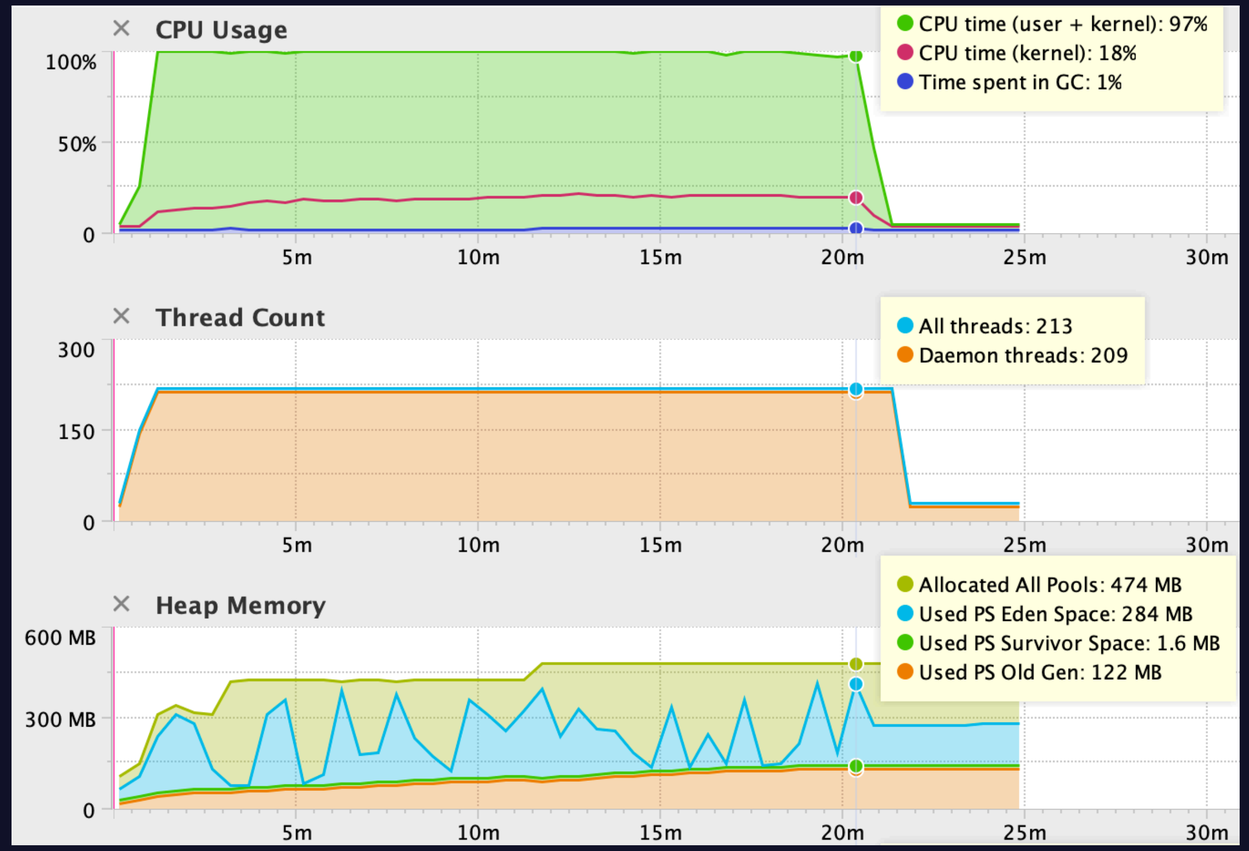
Spring MVC - Tomcat - 200 threads

With a simple hello world String rendering

 128 / steps up to 4000  20s  20min

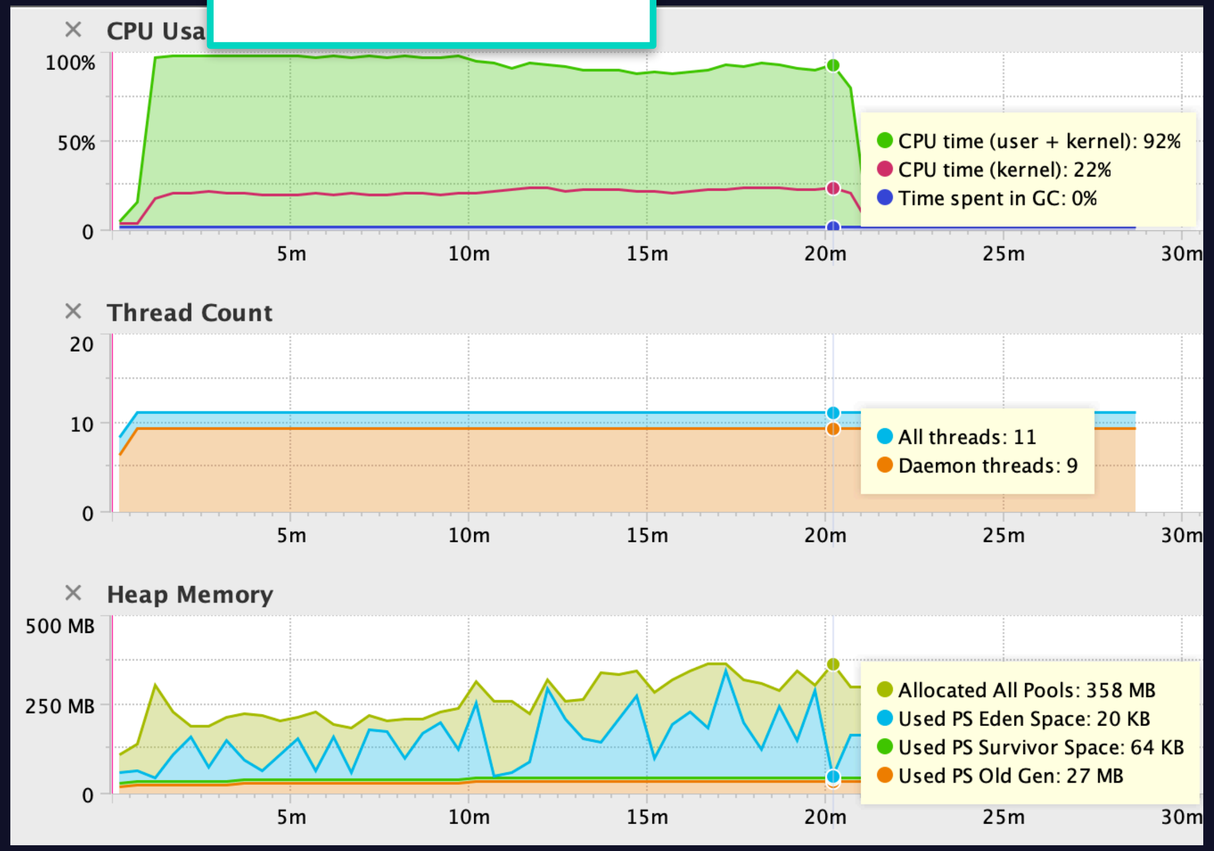
Order(s) of Magnitude more **efficient**



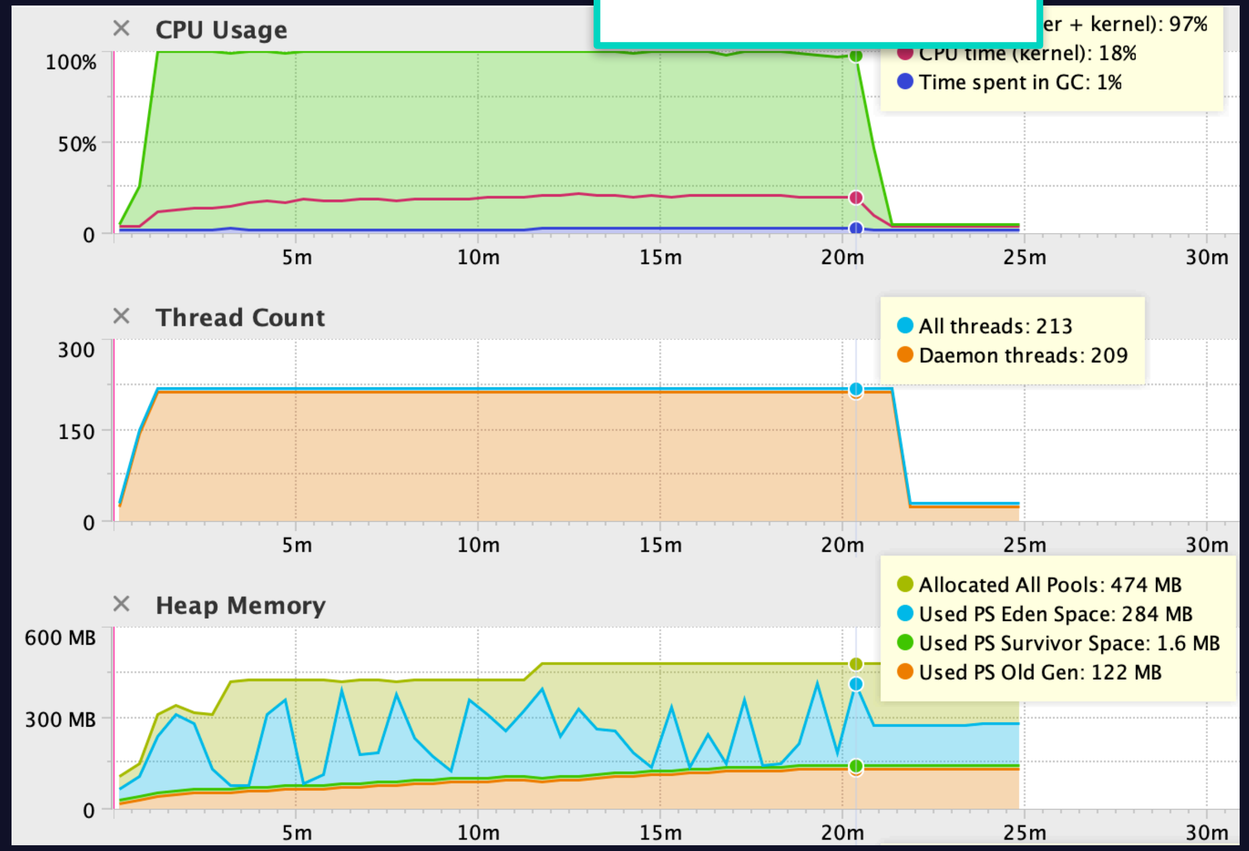


Order(s) of Magnitude more **efficient**

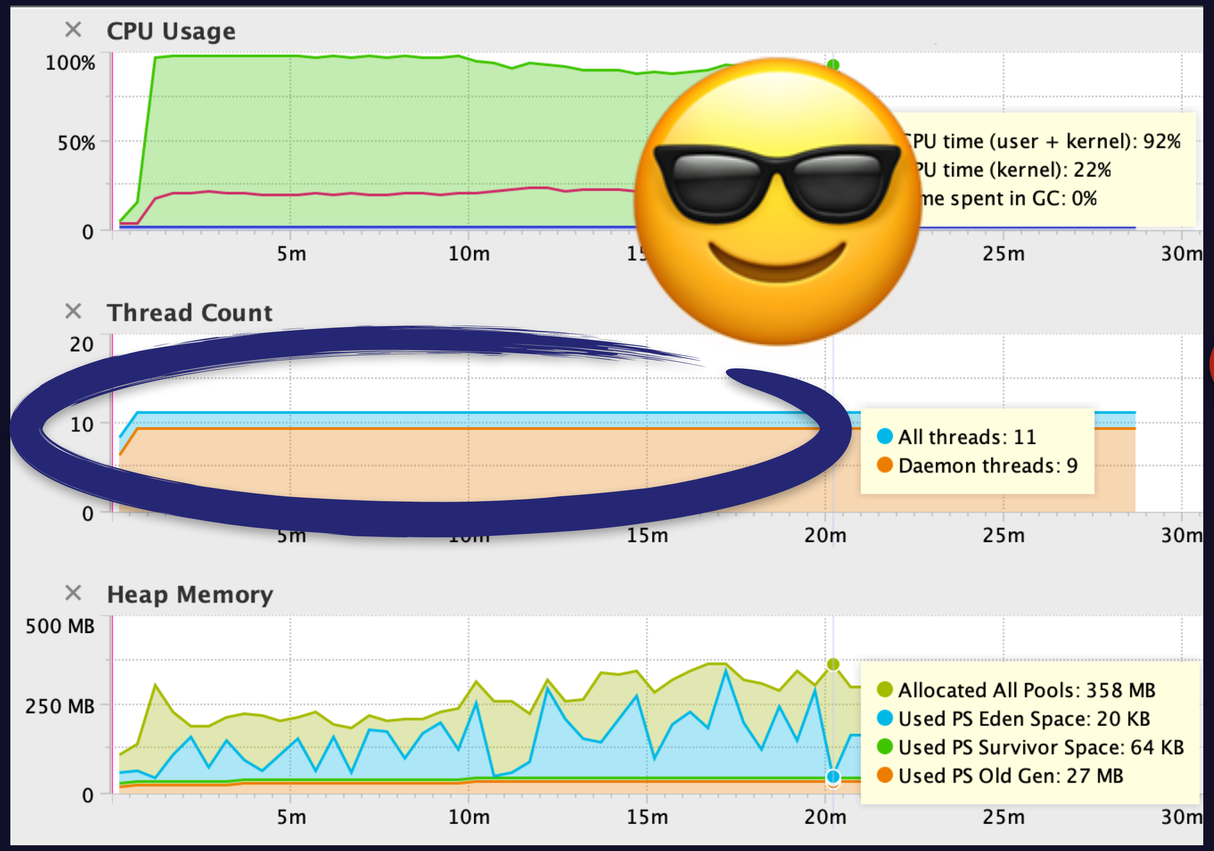
WebFlux

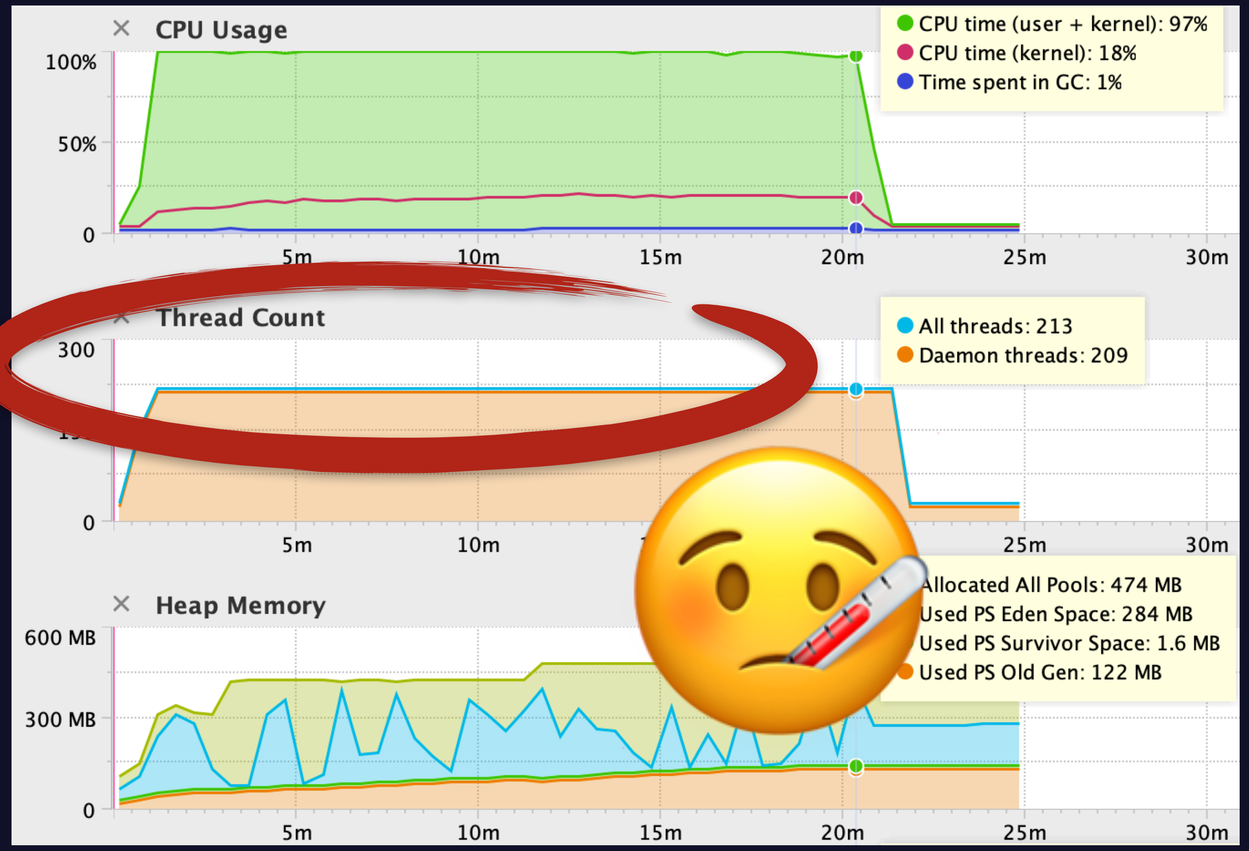


MVC

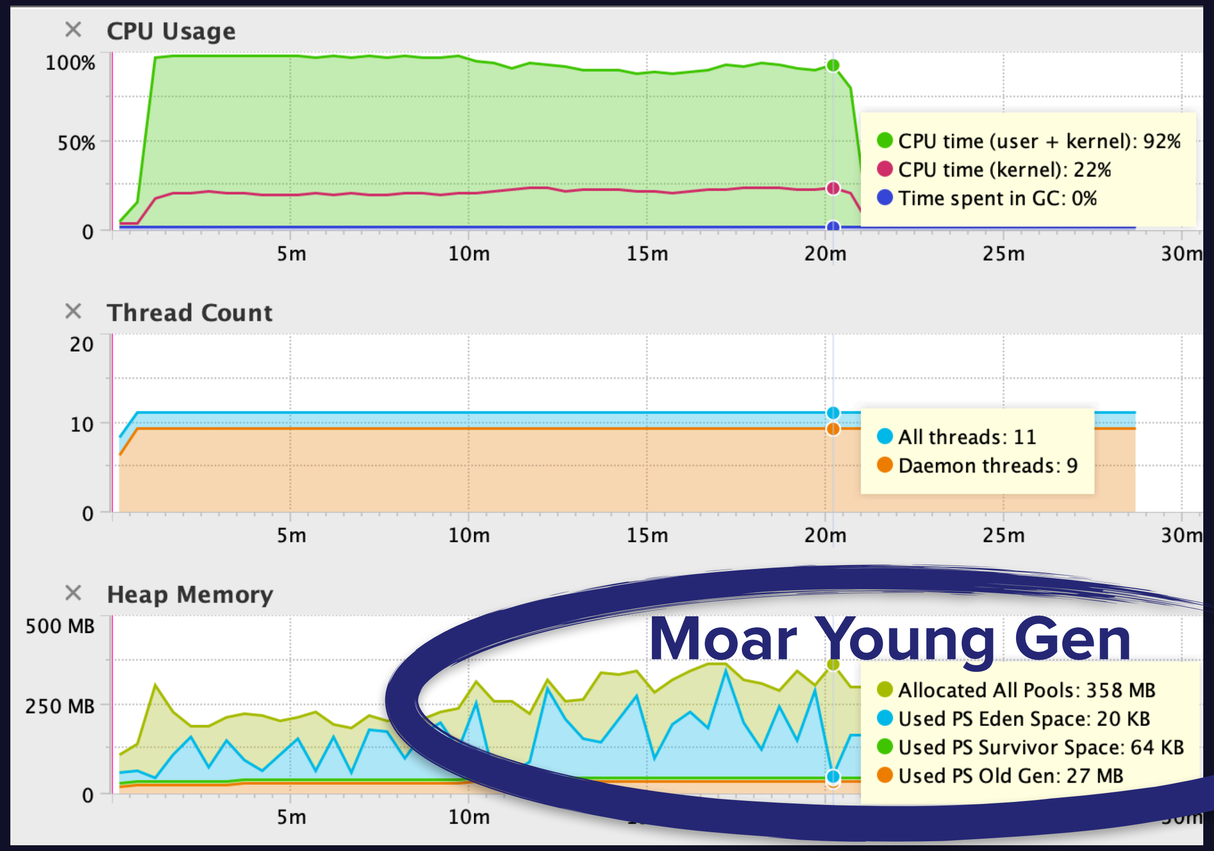


Order(s) of Magnitude more **efficient**

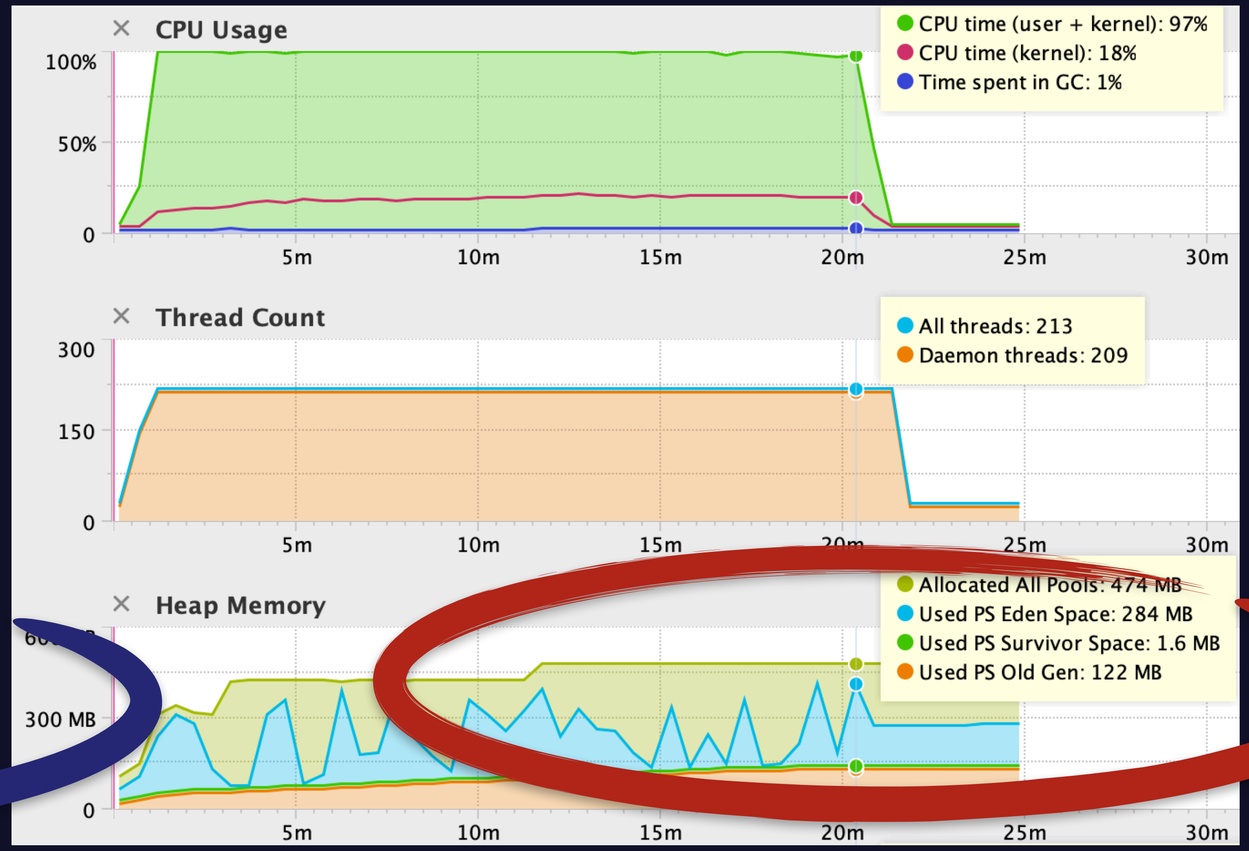




Order(s) of Magnitude more **efficient**

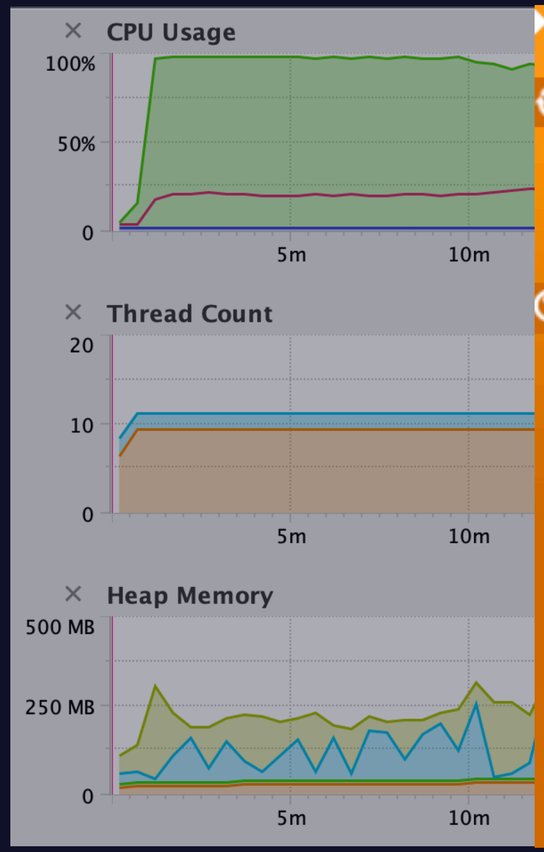


Moar Young Gen



Moar Old Gen

Order(s) of Magnitude more **efficient**



STATISTICS

Executions

	Total	OK	KO
	24340499	24340499	0
Mean req/s	20283.749	20283.749	-

Response Time (ms)

	Total	OK	KO
Min	0	0	-
50th percentile	123	123	-
75th percentile	171	171	-
95th percentile	227	226	-
99th percentile	318	318	-
Max	1398	1398	-
Mean	124	124	-
Std Deviation	68	68	-

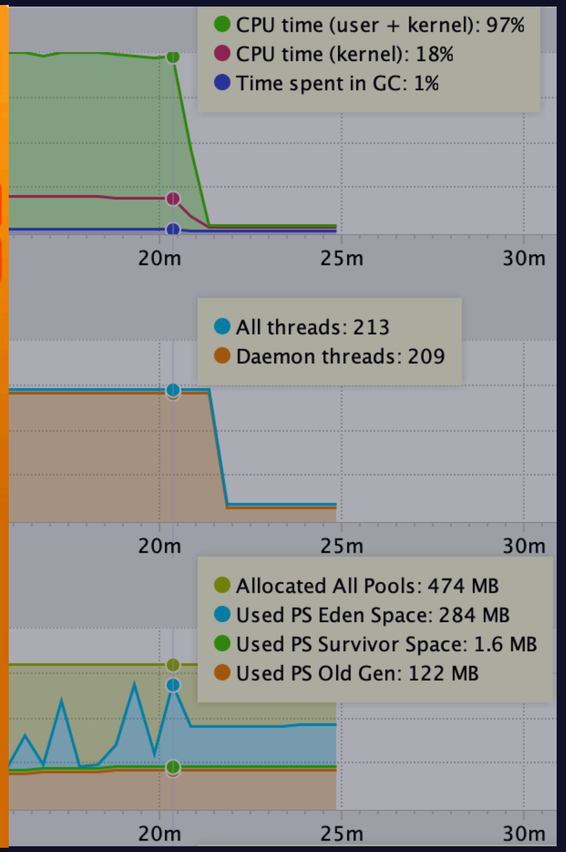
STATISTICS

Executions

	Total	OK	KO
	13865529	13865529	0
Mean req/s	11554.608	11554.608	-

Response Time (ms)

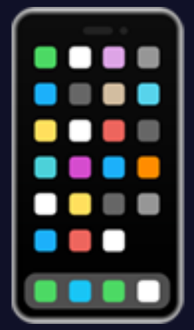
	Total	OK	KO
Min	0	0	-
50th percentile	203	203	-
75th percentile	298	298	-
95th percentile	455	455	-
99th percentile	610	610	-
Max	1797	1797	-
Mean	218	218	-
Std Deviation	130	130	-



Order(s) of Magnitude more **efficient**

Order(s) of Magnitude more **efficient**

Designed for **connection volume scalability**



Order(s) of Magnitude
more **efficient**

Designed for **connection volume scalability**

- **Lower** Number of instances running a same application
- **Lower** Impact from traffic latency diversity
- **Lower** Impact from persistent http traffic (websocket, sse, http2...)

Designed for **connection volume scalability**

Spring WebFlux.Fn - Reactor Netty - 4 threads

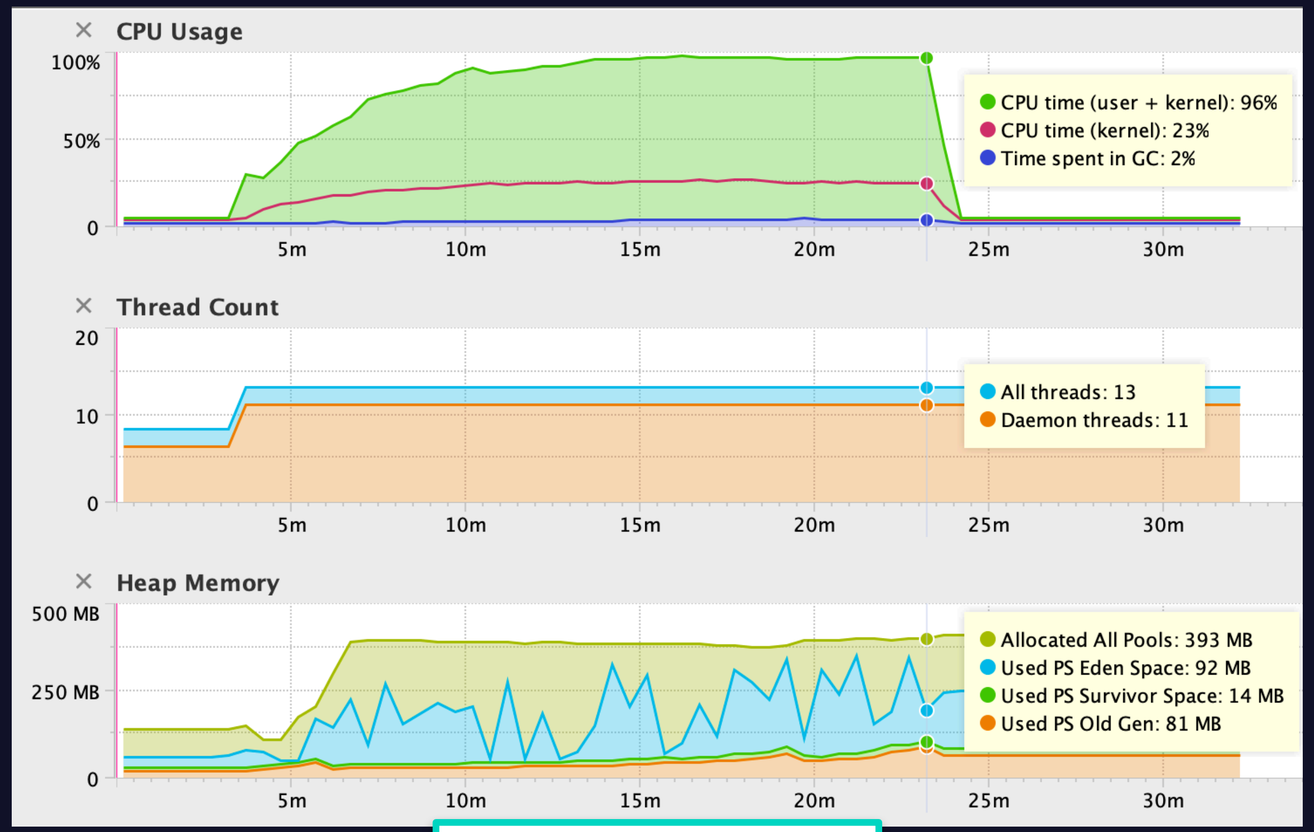
&

Spring MVC - Tomcat - 200 threads

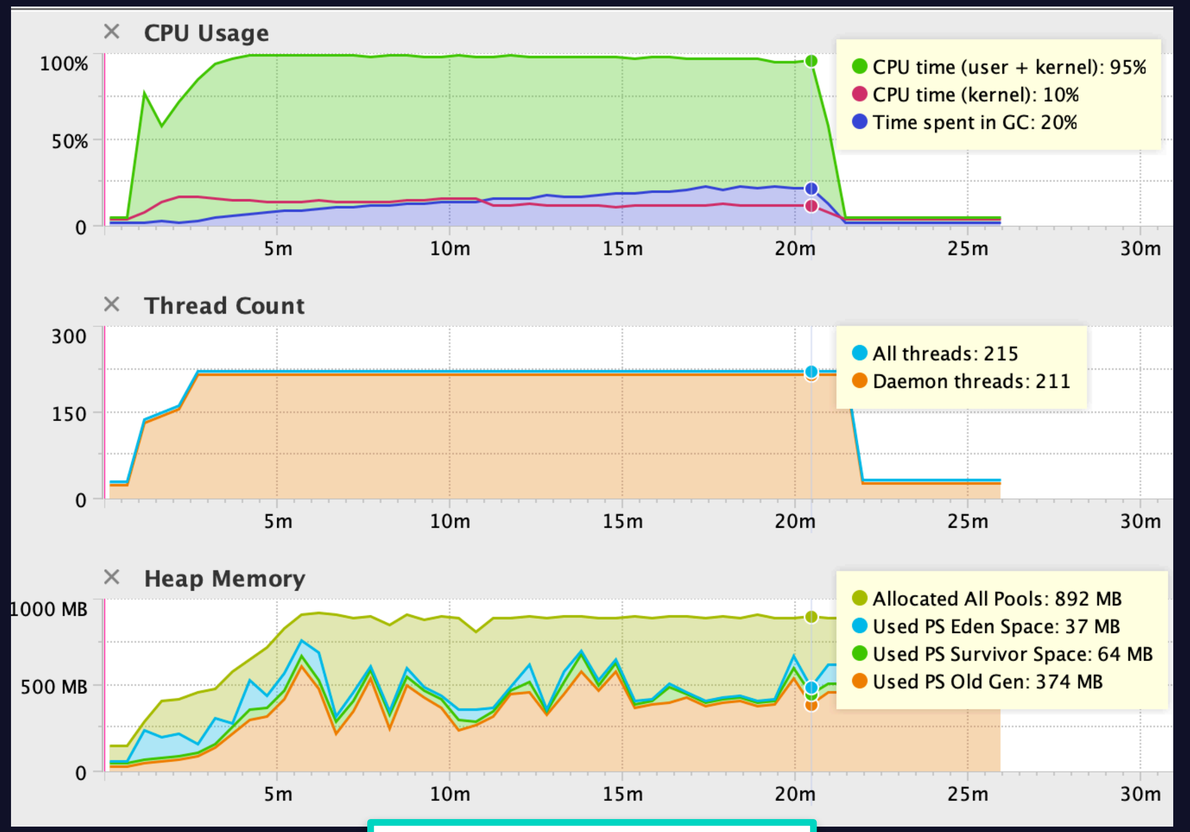
*With a **100ms delayed** hello world String rendering*

 128 / steps up to 4000  20s  20min

Designed for **connection volume scalability**

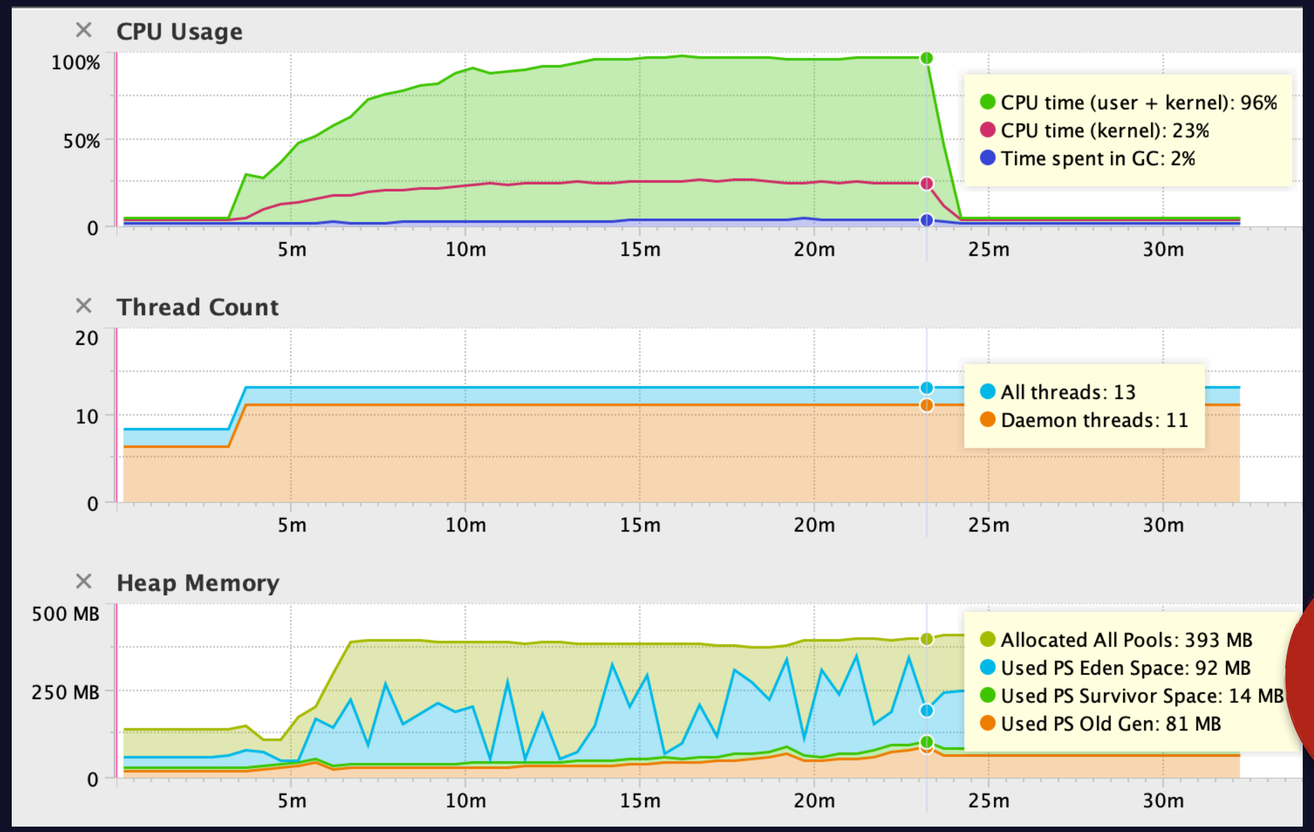


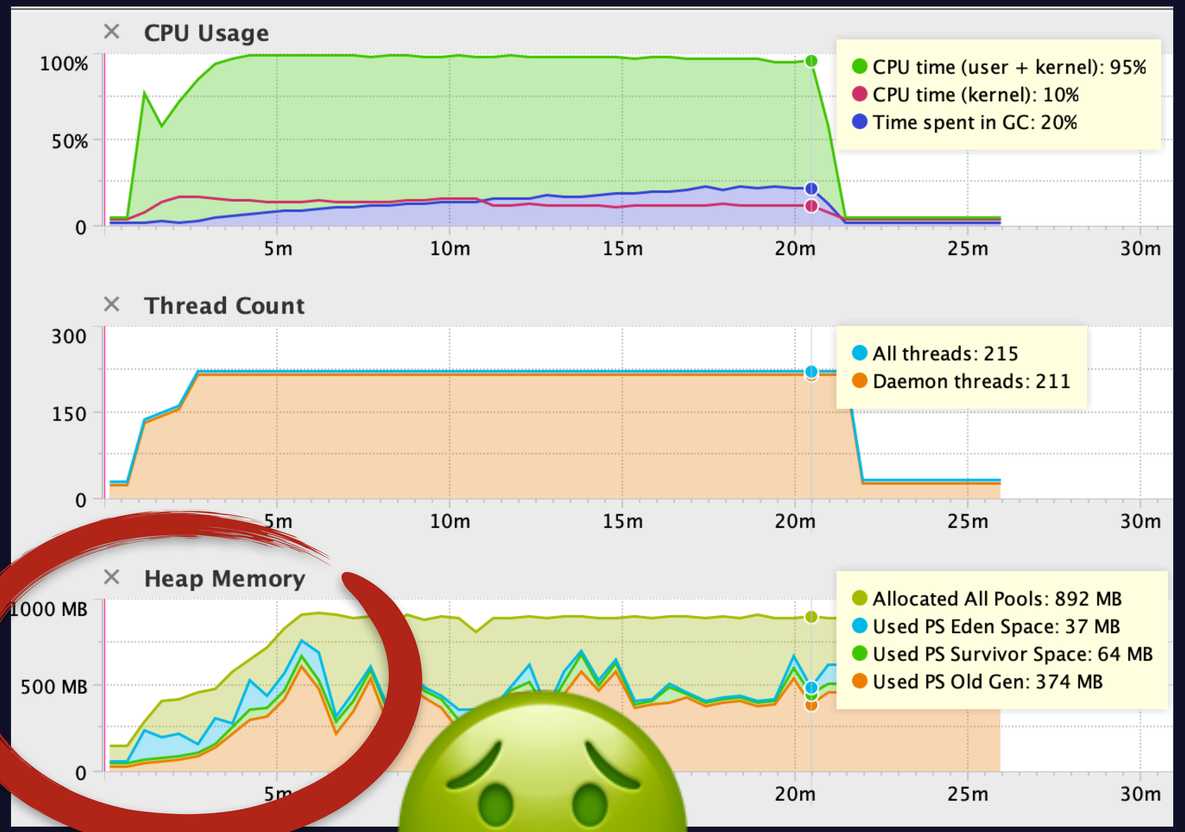
WebFlux

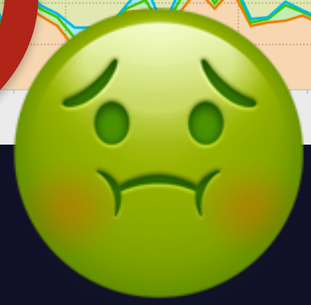


MVC (async)

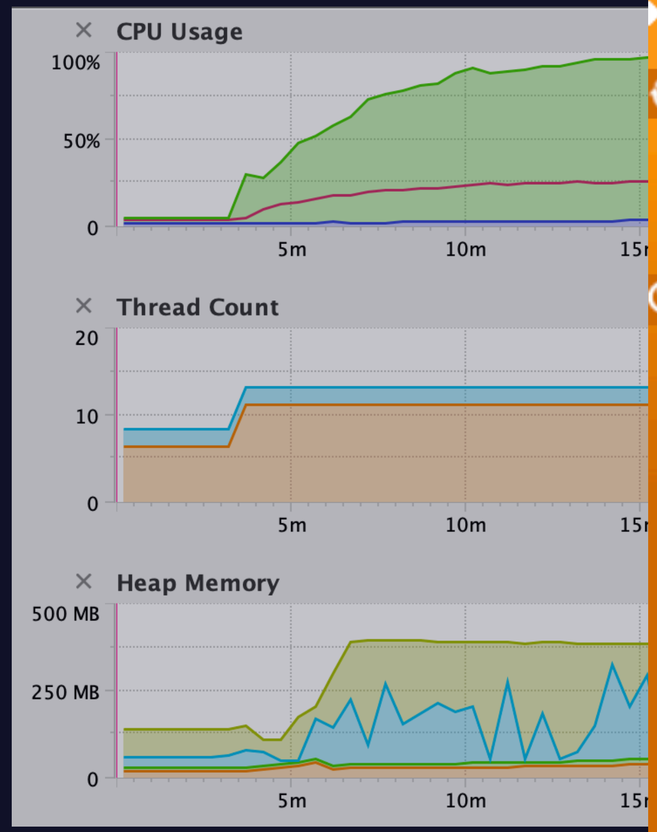
Designed for **connection volume scalability**







Designed for **connection volume scalability**



STATISTICS

Executions

	Total	OK	KO
	18920586	18920586	0
Mean req/s	15767.155	15767.155	-

Response Time (ms)

	Total	OK	KO
Min	100	100	-
50th percentile	157	157	-
75th percentile	189	189	-
95th percentile	243	243	-
99th percentile	310	310	-
Max	1201	1201	-
Mean	160	160	-
Std Deviation	49	49	-

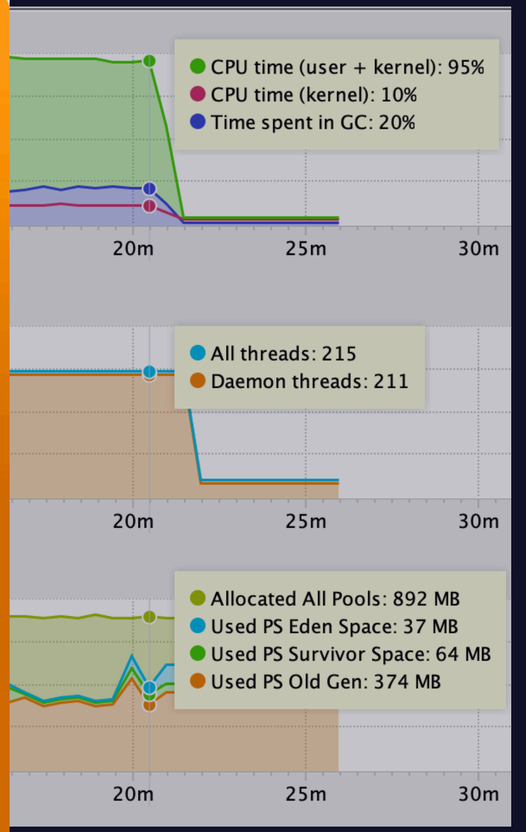
STATISTICS

Executions

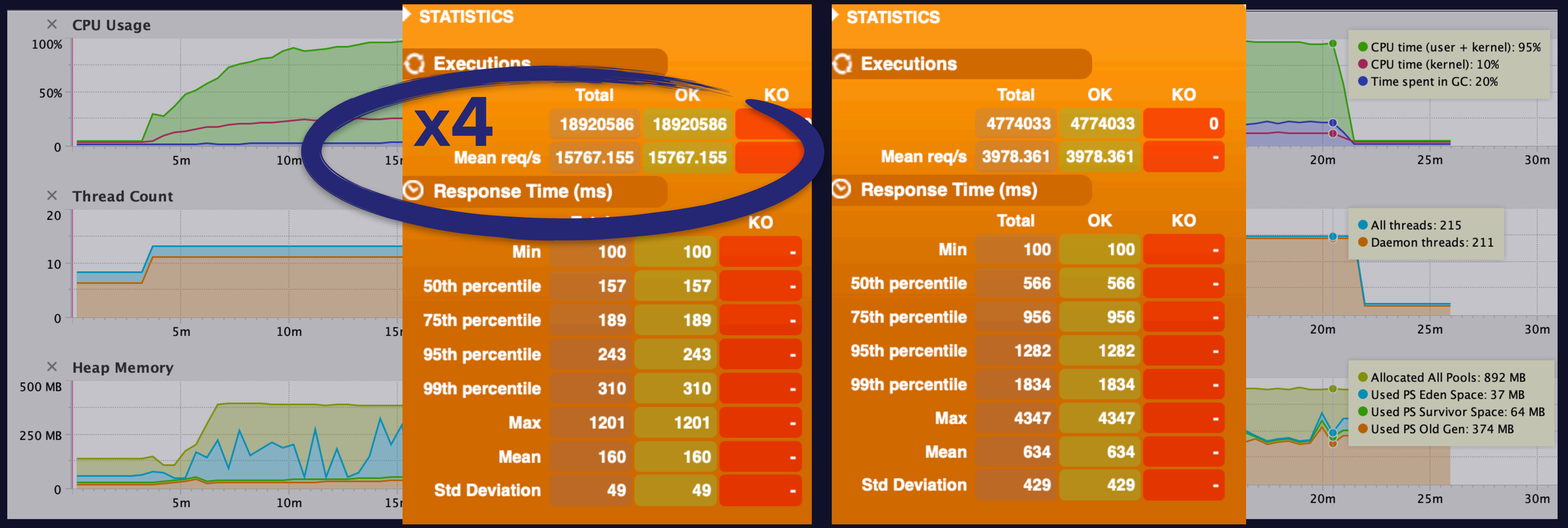
	Total	OK	KO
	4774033	4774033	0
Mean req/s	3978.361	3978.361	-

Response Time (ms)

	Total	OK	KO
Min	100	100	-
50th percentile	566	566	-
75th percentile	956	956	-
95th percentile	1282	1282	-
99th percentile	1834	1834	-
Max	4347	4347	-
Mean	634	634	-
Std Deviation	429	429	-



Designed for **connection volume scalability**



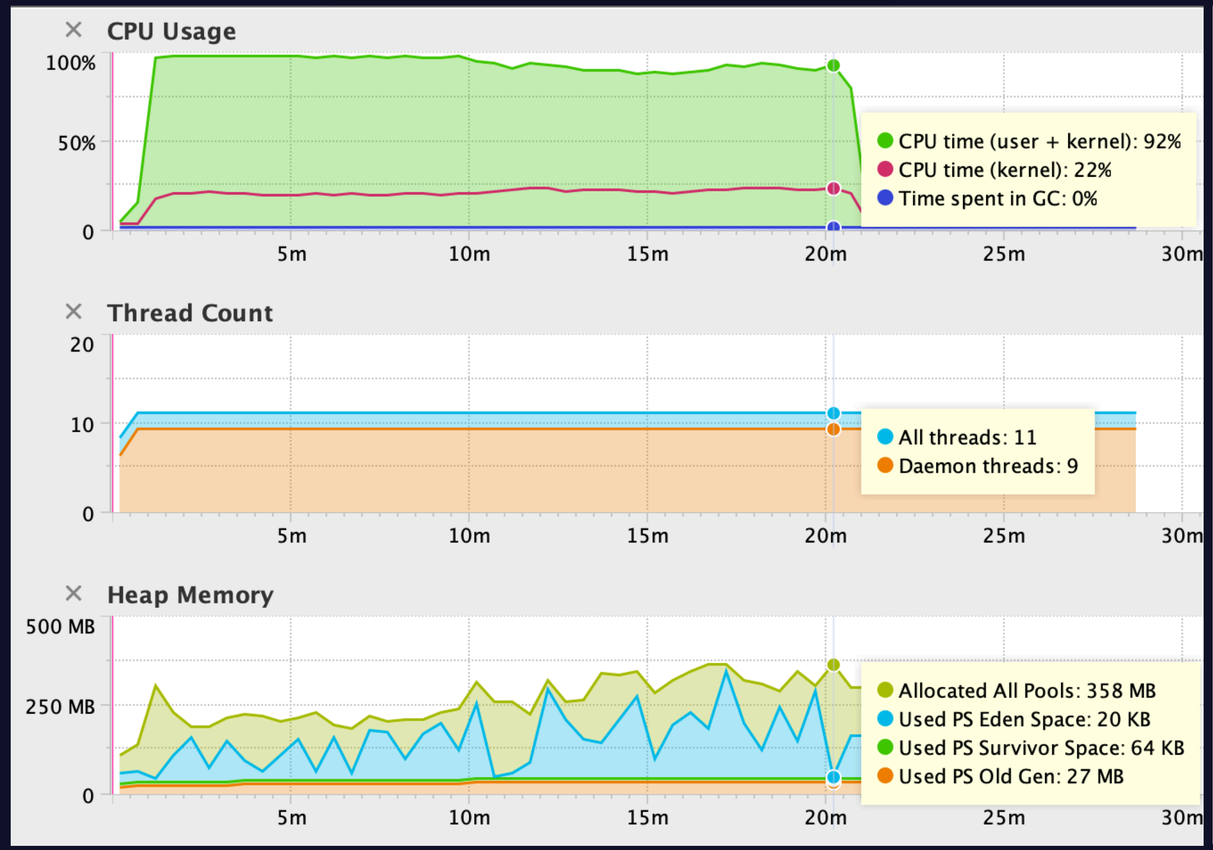
Designed for **connection volume scalability**

Designed for **connection volume scalability**

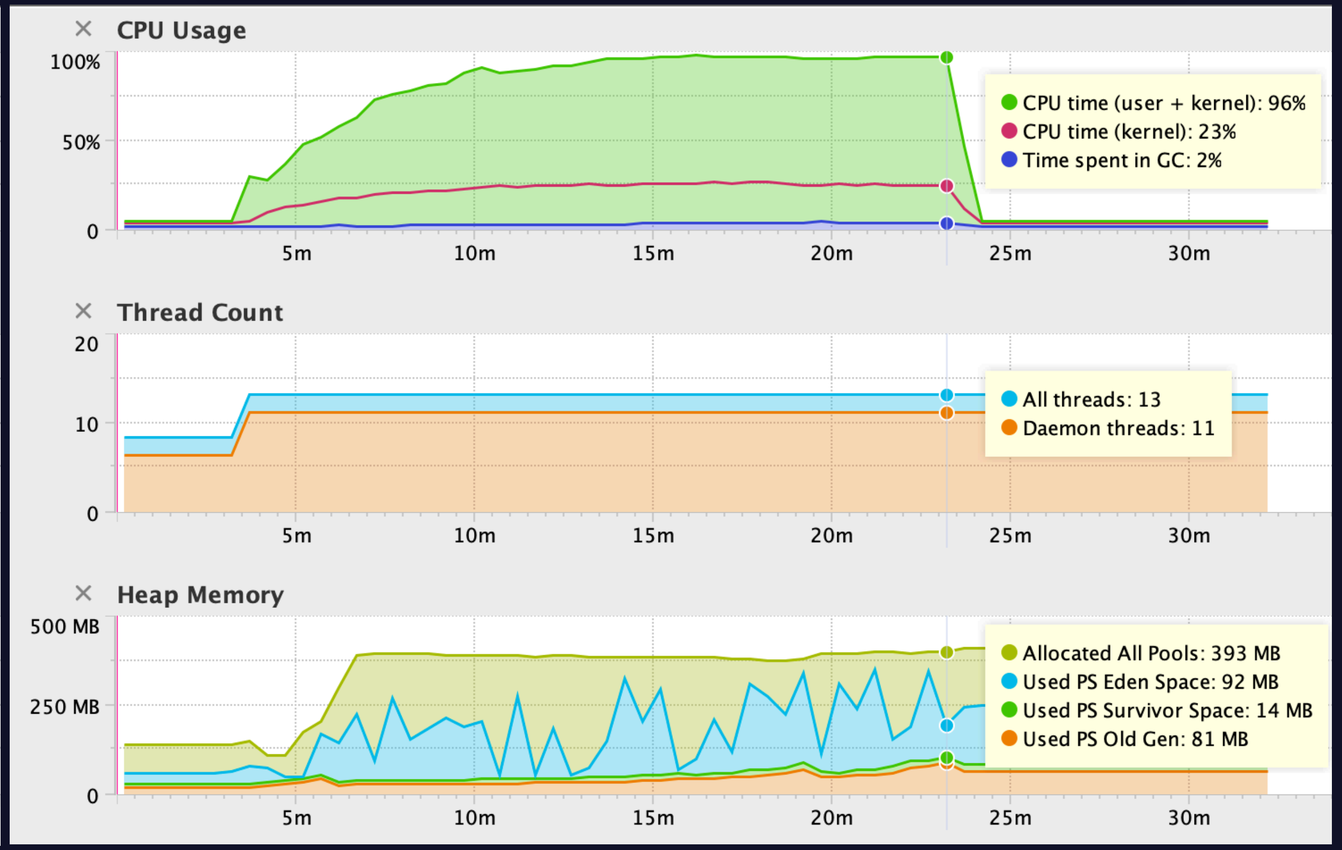
How does the response delay impact each model ?

Designed for **connection volume scalability**

WebFlux

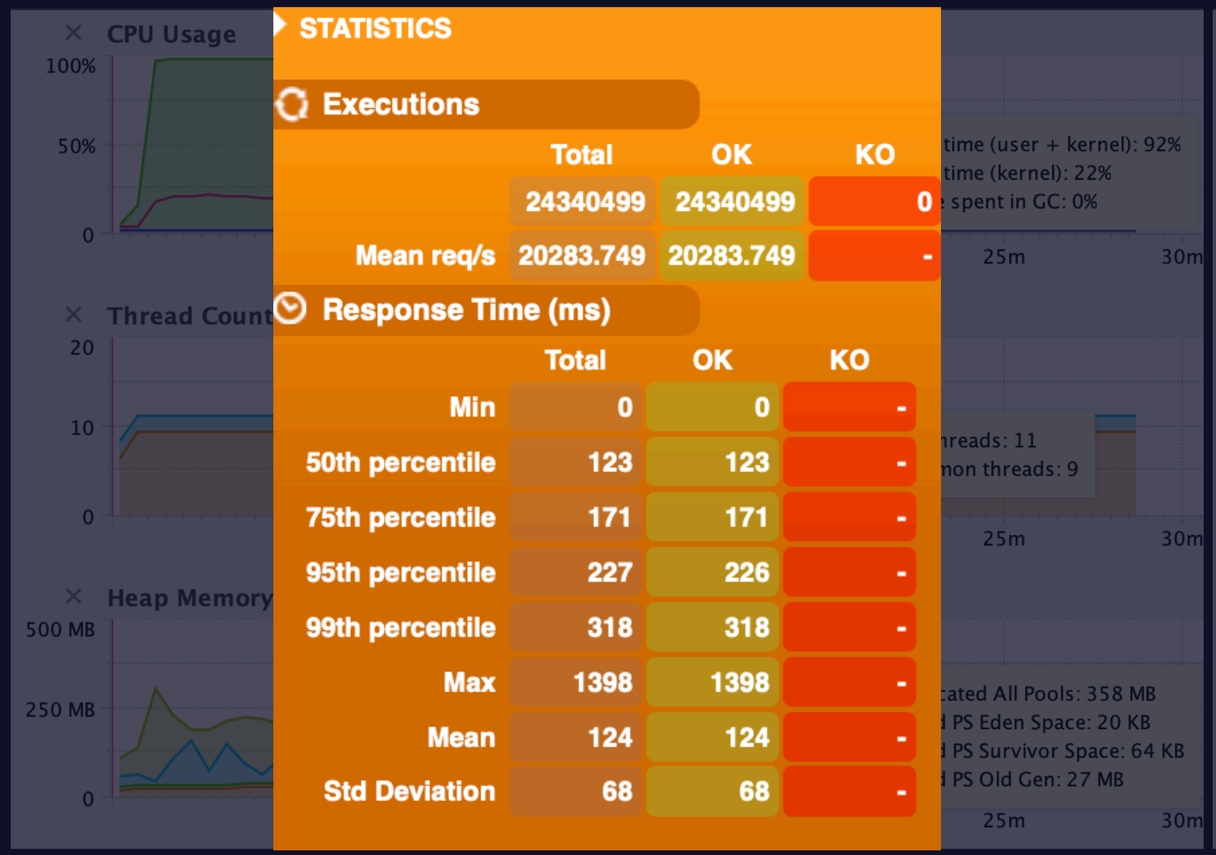


WebFlux (with response delay)

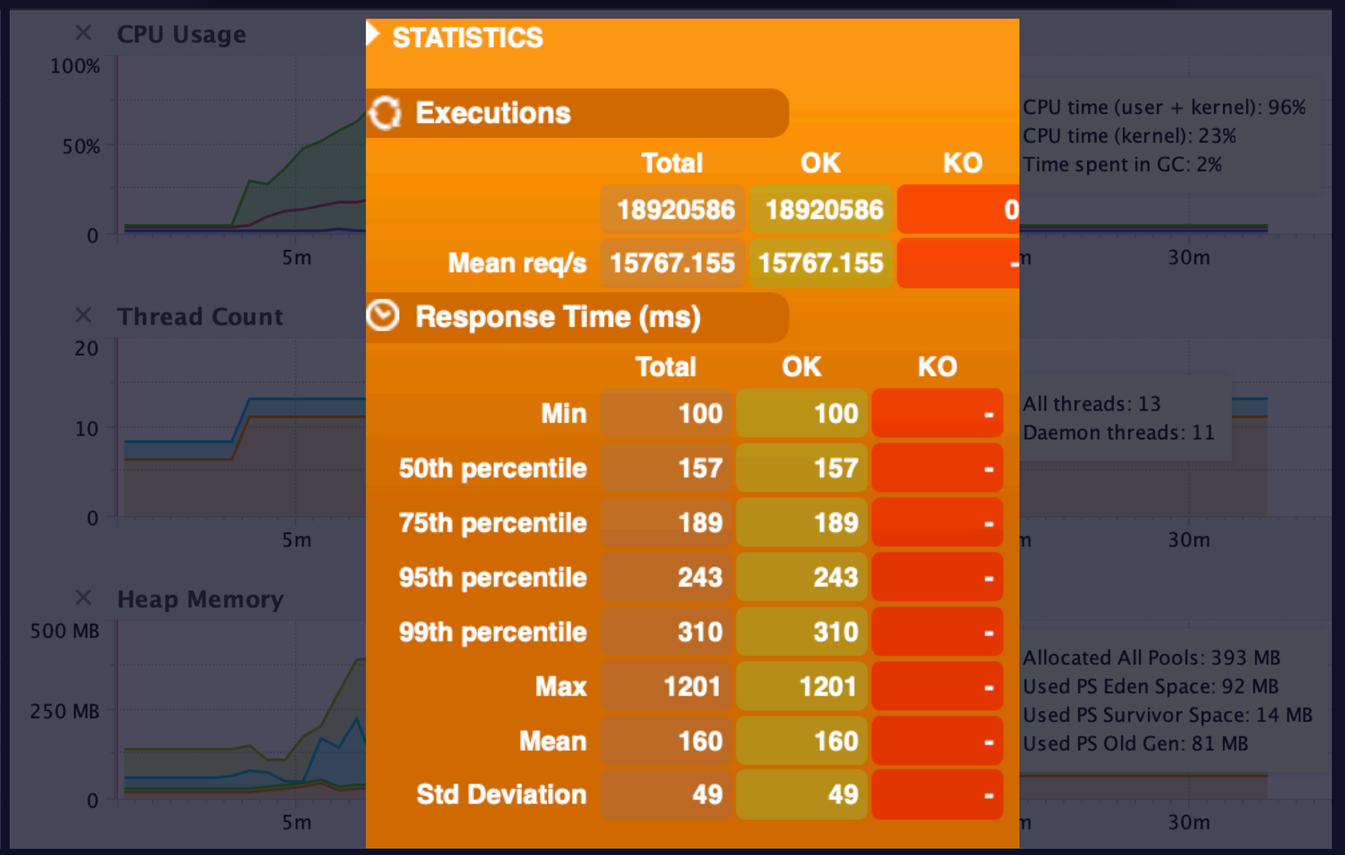


Designed for **connection volume scalability**

WebFlux



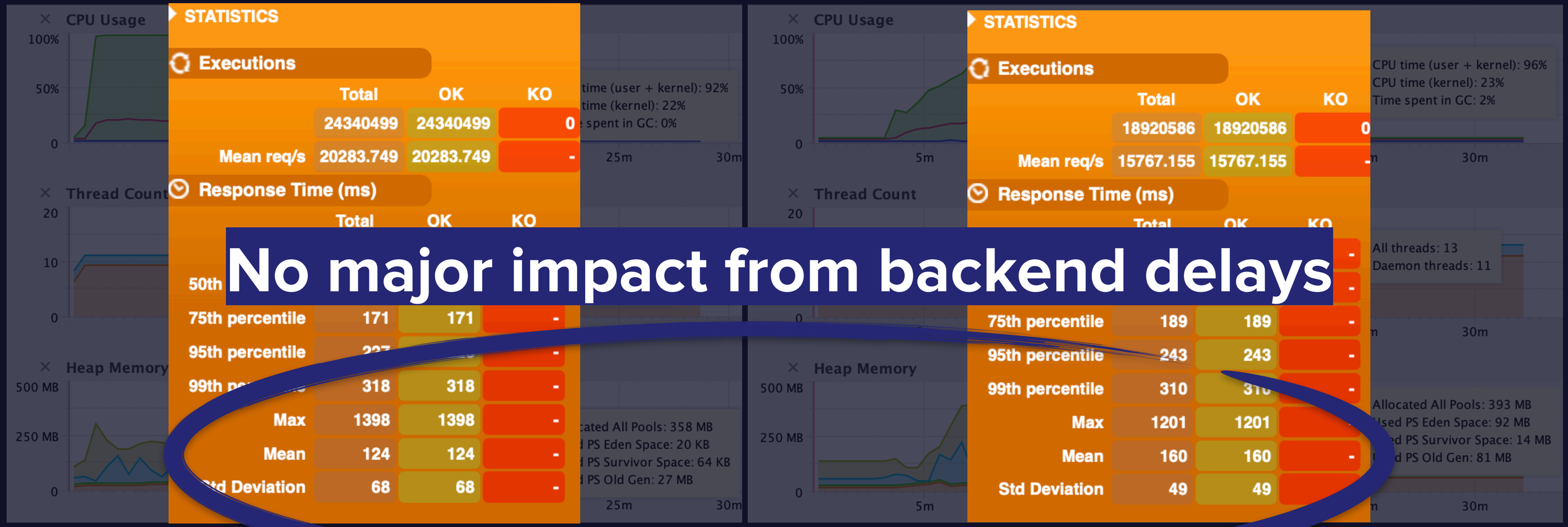
WebFlux (with response delay)



Designed for **connection volume scalability**

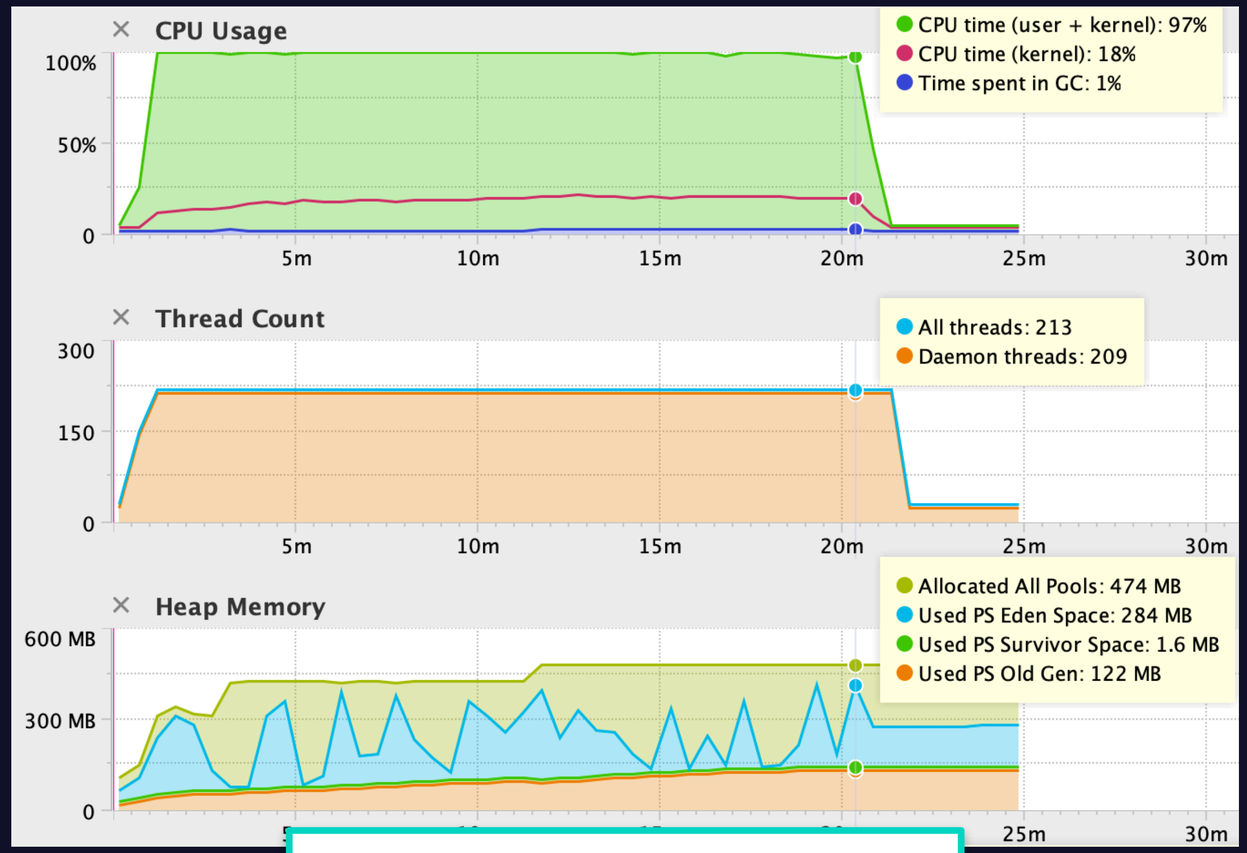
WebFlux

WebFlux (with response delay)

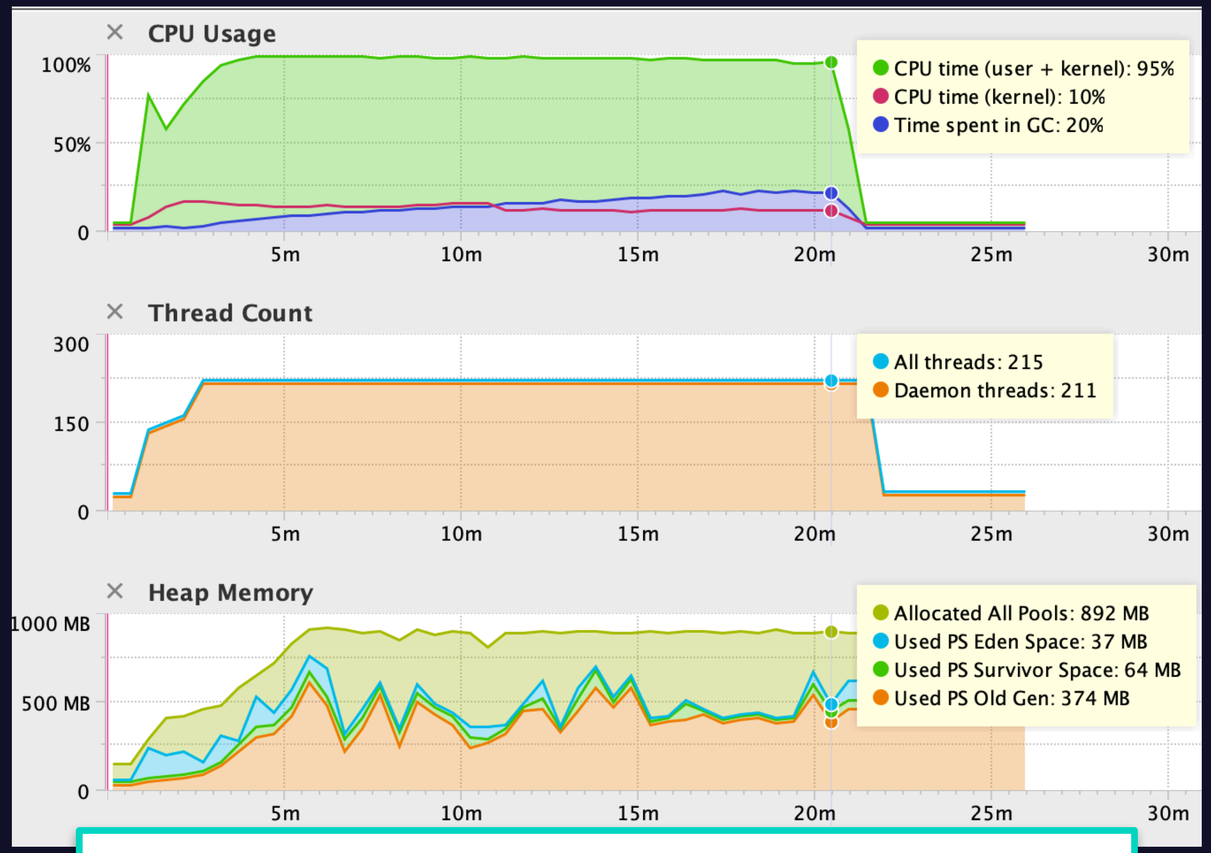


No major impact from backend delays

Designed for **connection volume scalability**

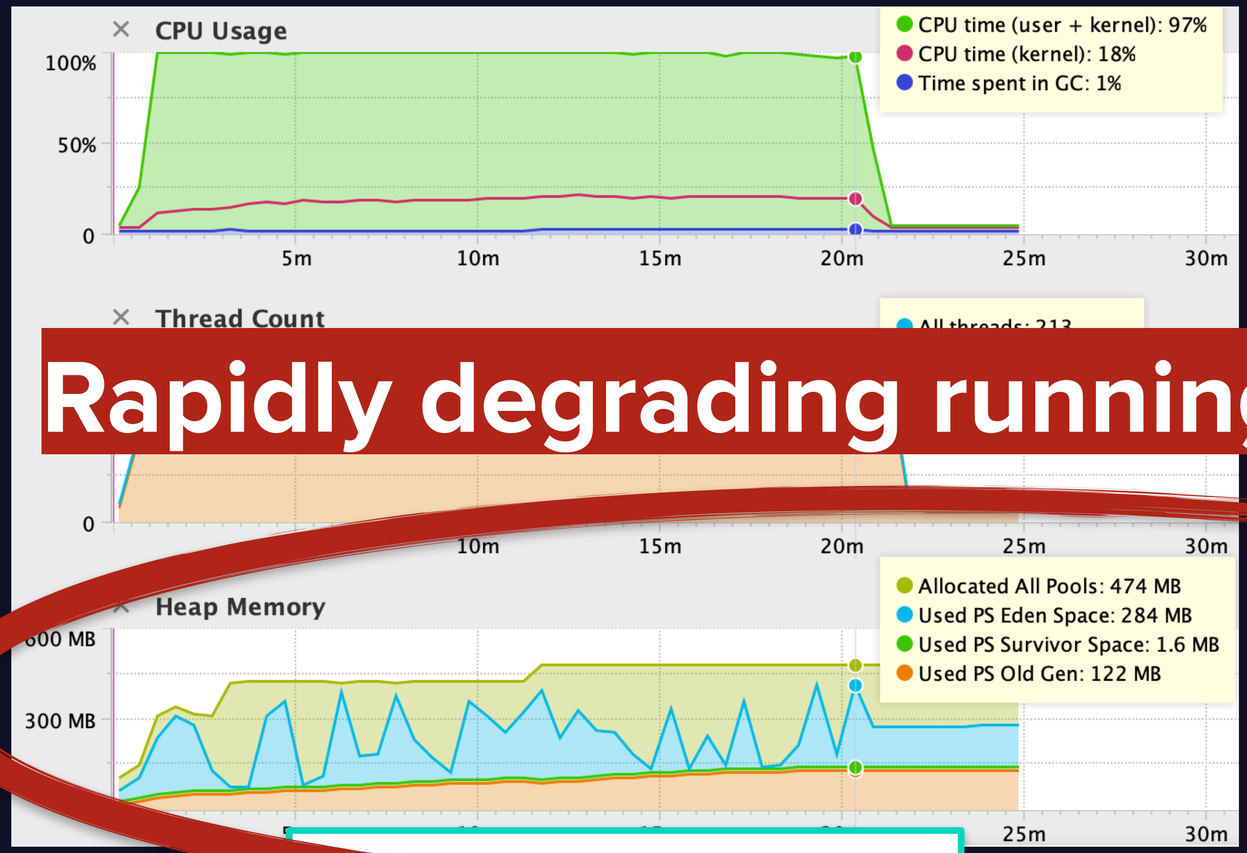


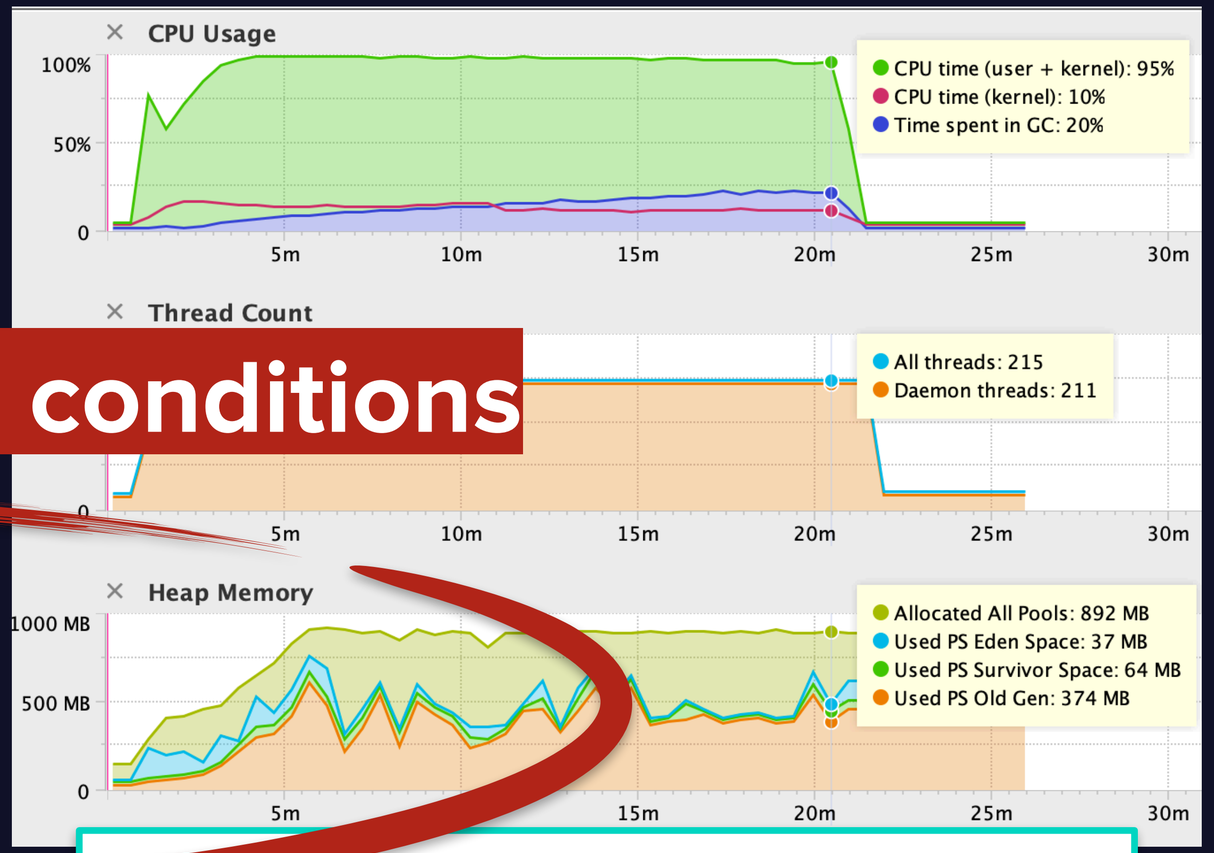
WebMVC



WebMVC (with response delay)

Designed for **connection volume scalability**



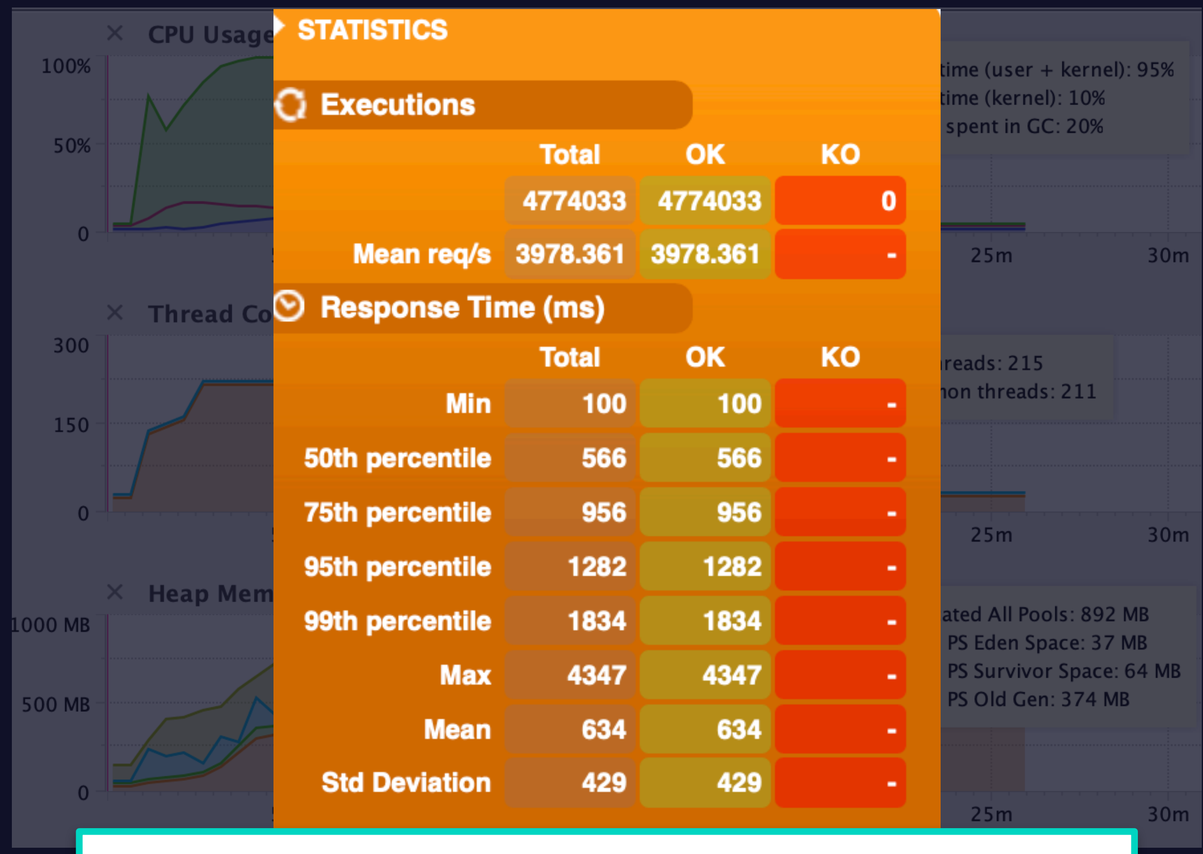


Rapidly degrading running conditions

WebMVC (with response delay)

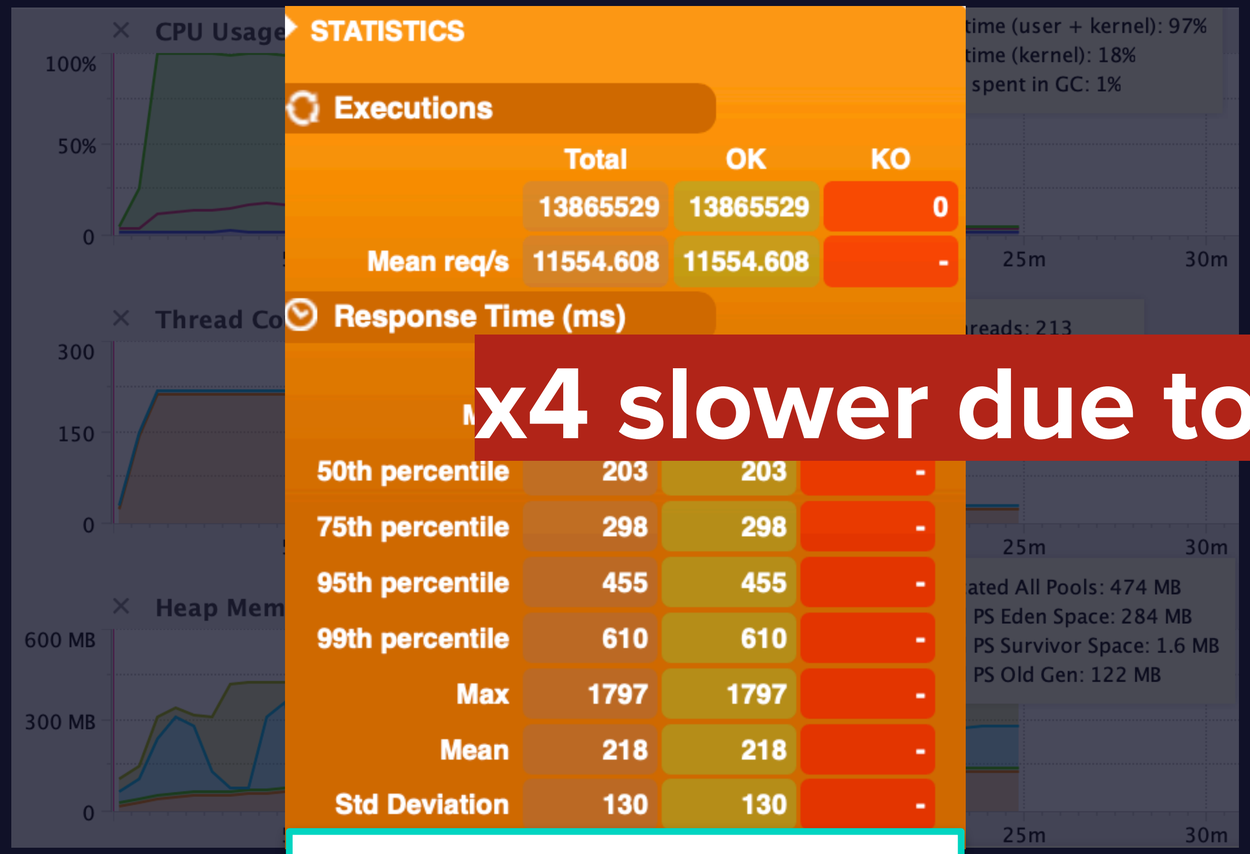
Designed for **connection volume scalability**

WebMVC

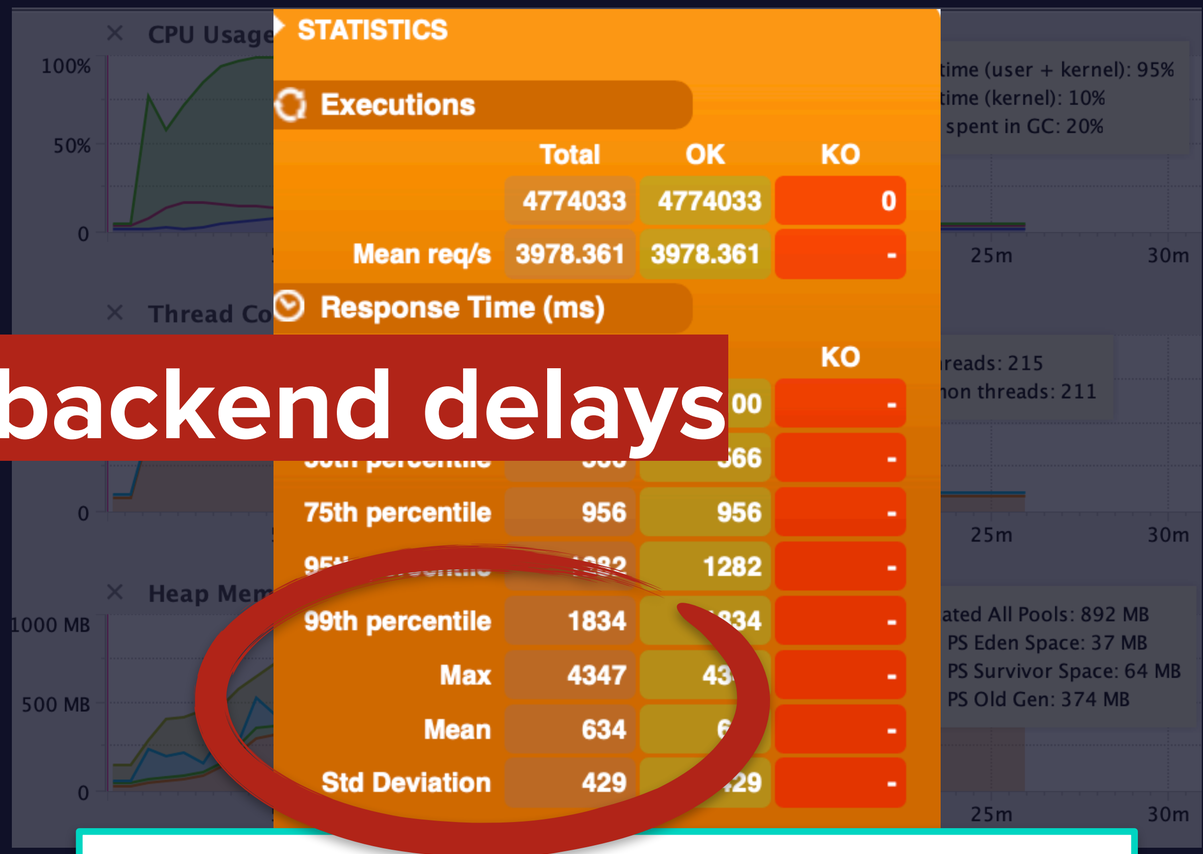


WebMVC (with response delay)

Designed for **connection volume scalability**



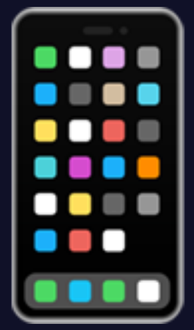
WebMVC



WebMVC (with response delay)

x4 slower due to backend delays

Designed for **connection volume scalability**



Order(s) of Magnitude
more **efficient**

Sleep Well,
get less paged

Order(s) of Magnitude
more **efficient**

Designed for
connection volume scalability

Sleep Well,
get less paged

- **Greater** resilience to runtime errors, including memory issues
- **Greater** availability at any point in time, specially under stress
- **Easier** resources use modeling

Sleep Well,
get less paged

Figure 1.a:
A blocking JVM HTTP server with no flow control

Sleep Well,
get less paged

Figure 1.a:
A blocking JVM HTTP server with no flow control

Sleep Well,
get less paged

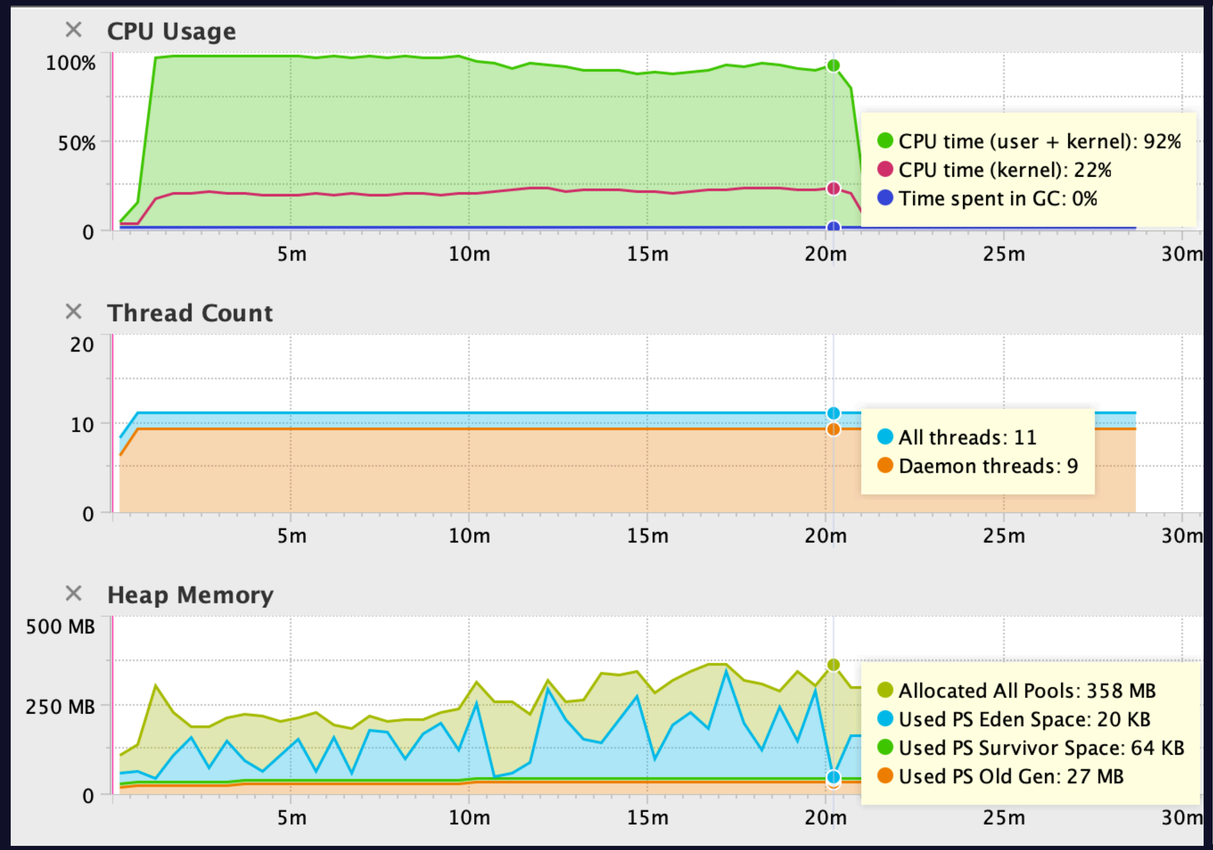
Reactive IO allows reading when application capacity permits

Sleep Well,
get less paged

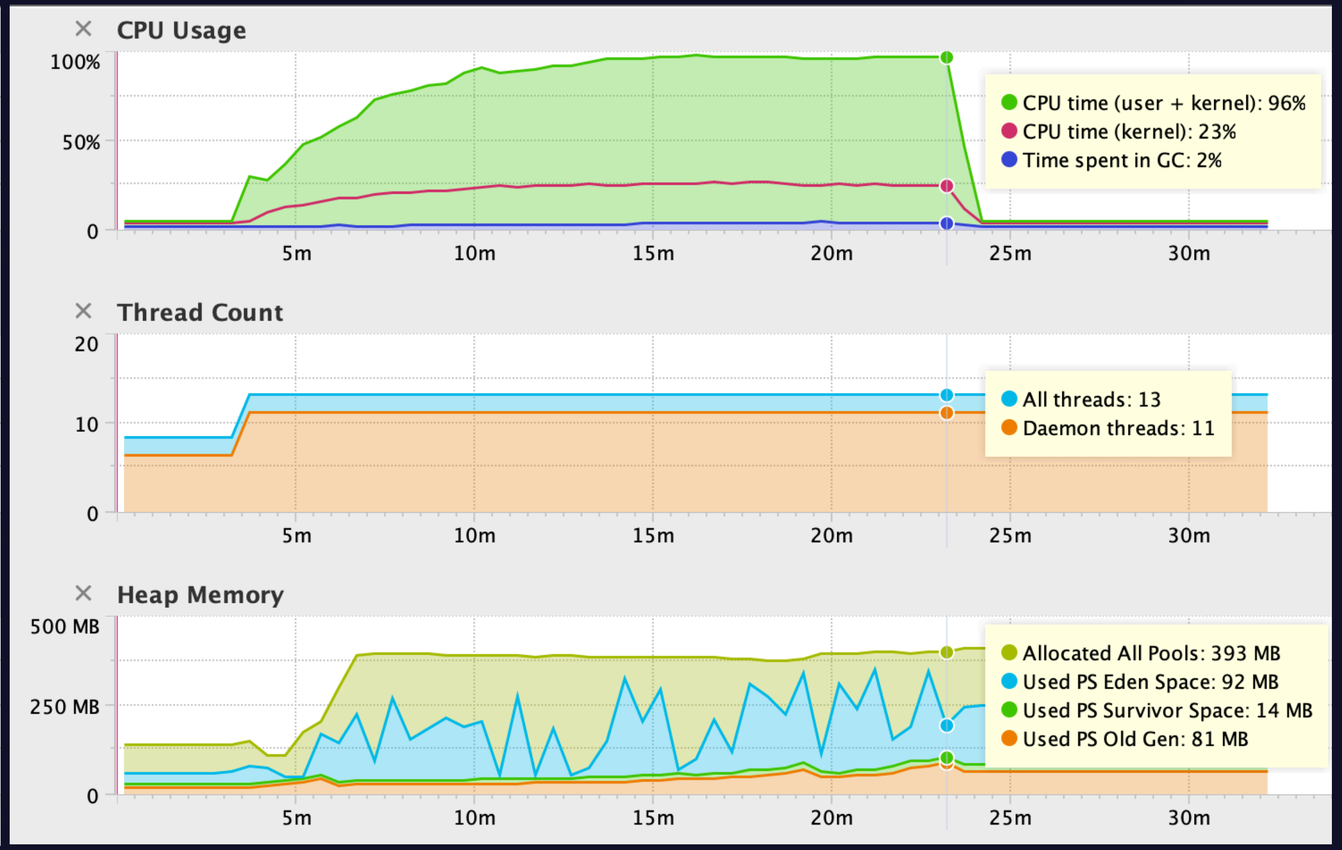
Reactive IO only requests application data to write when client capacity permits

Sleep Well, **get less paged**

WebFlux



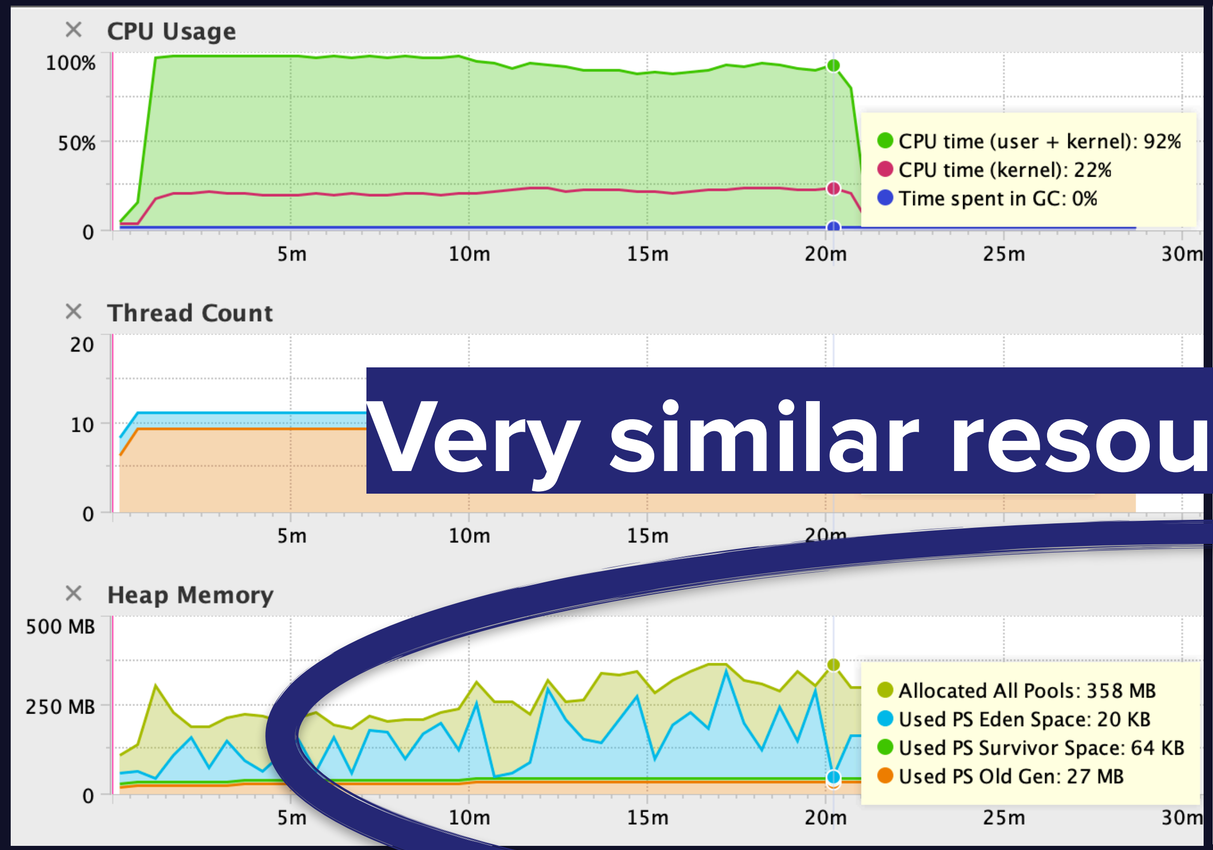
WebFlux (with response delay)

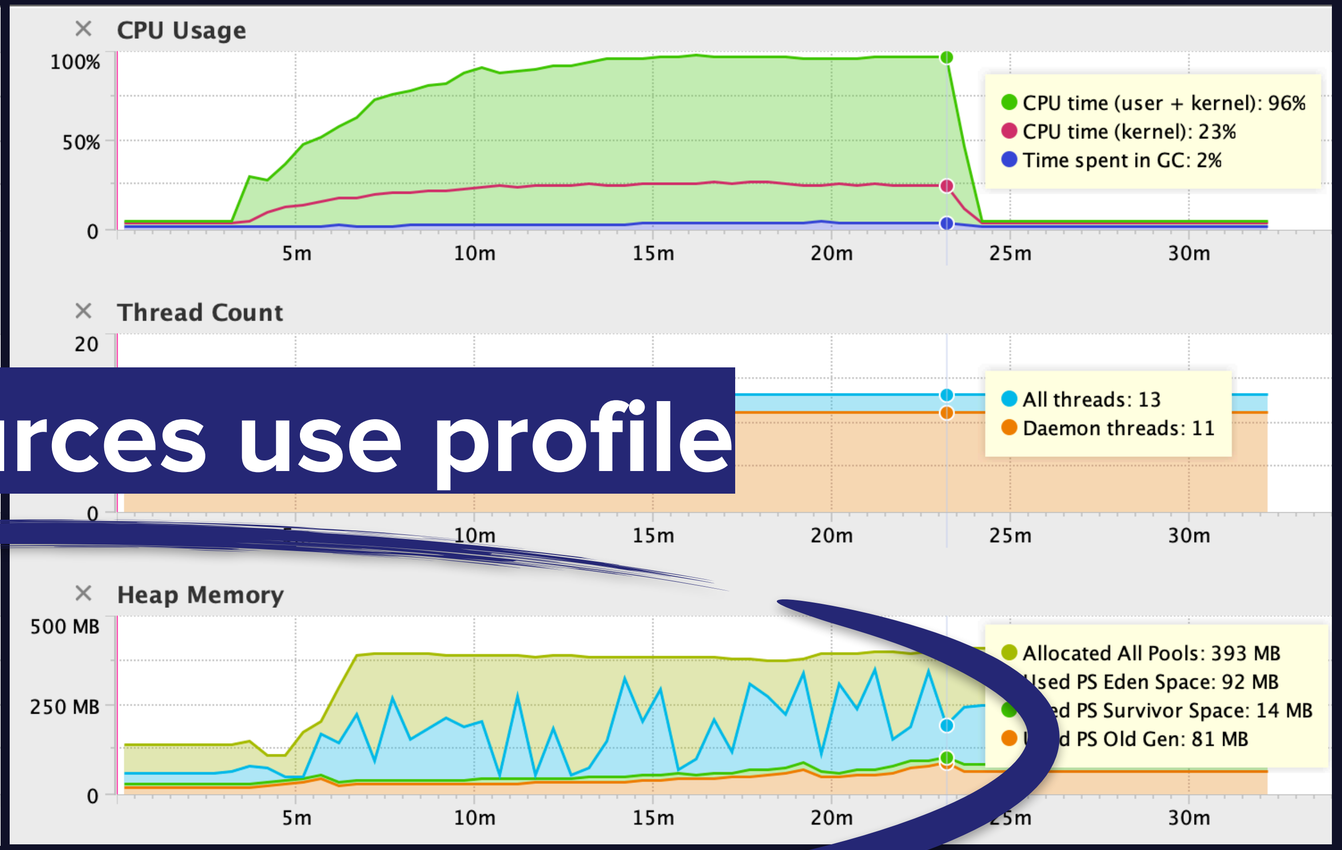


Sleep Well, **get less paged**

WebFlux

WebFlux (with response delay)





Very similar resources use profile

Sleep Well,
get less paged

Can reactive programming help with JVM OutOfMemory exceptions caused by incoming request traffic ?

Sleep Well,
get less paged

- **Yes**, using a reactive IO runtime, request body overflow is isolated
    - *Errors are propagated via the reactive flow error route : onError*
    - *Associated connection will usually be closed or a fallback route will be provided*

- With a blocking runtime, errors are bubbling up and OutOfMemory will be fatal
    - *Long JVM pauses will lead to a full app crash or unrecoverable state*

Order(s) of Magnitude
more **efficient**

Designed for
connection volume scalability

Sleep Well,
get less paged

Order(s) of Magnitude
more **efficient**

Designed for
connection volume scalability

Sleep Well,
get less paged

Event-Driven

Less Threads - Memory

Start quicker

Improve Latency*

Unlocks hyper concurrency

Connected Experience

Slow/Fast traffic

Near-Always Available

Flow Control

Predictable load

Resilient

What do you do with these qualities ?

Write Any Microservice

Resilient
Smaller resources footprint
Remote calls parallelization
Hedge Clients

Mobile Backend
Scale even with different traffic latency
Connected experience (notifications...)
Volume of clients

Edge API
Available
Traffic control
Volume of Connections

Write Any Microservice

Resilient

Smaller resources footprint

Remote calls parallelization

Hedge Clients

Mobile Backend
Scale even with different traffic latency
Connected experience (notifications...)
Volume of clients

Edge API
Available
Traffic control
Volume of Connections

Remote calls parallelization

With Spring, you can start writing your app with **WebMVC**,

introduce **WebClient** to improve backend calls workflows,

then consider moving to **WebFlux**

Mobile Backend
Scale even with different traffic latency
Connected experience (notifications...)
Volume of clients

Edge API
Available
Traffic control
Volume of Connections

Write an Edge API

Available
Traffic Control
Volume of Connections

Mobile Backend
Scale even with different traffic latency
Connected experience (notifications...)
Volume of clients

Write an Edge API

Available

Traffic Control

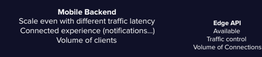
Volume of Connections

Mobile Backend
Scale even with different traffic latency
Connected experience (notifications...)
Volume of clients

Traffic Control

A reactive runtime will isolate ingress/egress traffic latency and exceptions.

Given this isolation property, a reactive Edge API should always be able to route traffic, even limit it or short-circuit when required.



Mobile Backend

Scale with different traffic latency
Connected experience (notifications...)
Volume of clients

~~Mobile~~ IoT Backend

Scale with different traffic latency
Connected experience (notifications...)
Volume of clients

Ok sounds cool, but what about the developer experience ?

Exceptions since Spring Framework 5.2

```
Error has been observed at the following site(s):
    |_ checkpoint ⇢ Handler io.spring.workshop.tradingservice.QuotesController#quotesDetails(String) [DispatcherHandler]
    |_ checkpoint ⇢ org.springframework.security.web.server.authorization.AuthorizationWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.authorization.ExceptionTranslationWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.authentication.logout.LogoutWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.savedrequest.ServerRequestCacheWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.context.SecurityContextServerWebExchangeWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.ui.LogoutPageGeneratingWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.ui.LoginPageGeneratingWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.authentication.AuthenticationWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.authentication.AuthenticationWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.context.ReactorContextWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.csrf.CsrfWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.header.HttpHeaderWriterWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.config.web.server.ServerHttpSecurity$ServerWebExchangeReactorContextWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.WebFilterChainProxy [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.boot.web.reactive.filter.OrderedHiddenHttpMethodFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.boot.actuate.metrics.web.reactive.server.MetricsWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ HTTP GET "/quotes/summary/MSFT" [ExceptionHandlingWebHandler]
```

Exceptions since Spring Framework 5.2

```
Error has been observed at the following site(s):
    |_ checkpoint ⇢ Handler io.spring.workshop.tradingservice.QuotesController#quotesDetails(String) [DispatcherHandler]
    |_ checkpoint ⇢ org.springframework.security.web.server.authorization.AuthorizationWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.authorization.ExceptionTranslationWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.authentication.logout.LogoutWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.savedrequest.ServerRequestCacheWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.context.SecurityContextServerWebExchangeWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.ui.LogoutPageGeneratingWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.ui.LoginPageGeneratingWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.authentication.AuthenticationWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.authentication.AuthenticationWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.context.ReactorContextWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.csrf.CsrfWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.header.HttpHeaderWriterWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.config.web.server.ServerHttpSecurity$ServerWebExchangeReactorContextWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.security.web.server.WebFilterChainProxy [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.boot.web.reactive.filter.OrderedHiddenHttpMethodFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ org.springframework.boot.actuate.metrics.web.reactive.server.MetricsWebFilter [DefaultWebFilterChain]
    |_ checkpoint ⇢ HTTP GET "/quotes/summary/MSFT" [ExceptionHandlingWebHandler]
```

Actionable stacktraces

Exceptions with debug mode enabled

```
Error has been observed at the following site(s):
    |_                Mono.doOnNext ⇢ at io.spring.workshop.tradingservice.TradingCompanyClient.getTradingCompany(TradingCompanyClient.java:38)
    |_            Mono.switchIfEmpty ⇢ at io.spring.workshop.tradingservice.TradingCompanyClient.getTradingCompany(TradingCompanyClient.java:39)
    |_                 Mono.zipWith ⇢ at io.spring.workshop.tradingservice.QuotesController.quotesDetails(QuotesController.java:39)
    |_            Mono.switchIfEmpty ⇢ at org.springframework.http.codec.EncoderHttpMessageWriter.write(EncoderHttpMessageWriter.java:119)
    |_                 Mono.flatMap ⇢ at org.springframework.http.codec.EncoderHttpMessageWriter.write(EncoderHttpMessageWriter.java:123)
    |_                   checkpoint ⇢ Handler io.spring.workshop.tradingservice.QuotesController#quotesDetails(String) [DispatcherHandler]
    |_                 Mono.flatMap ⇢ at org.springframework.web.reactive.DispatcherHandler.lambda$handleResult$5(DispatcherHandler.java:172)
    |_            Mono.onErrorResume ⇢ at org.springframework.web.reactive.DispatcherHandler.handleResult(DispatcherHandler.java:171)
    |_                 Mono.flatMap ⇢ at org.springframework.web.reactive.DispatcherHandler.handle(DispatcherHandler.java:147)
    |_                   Mono.defer ⇢ at org.springframework.web.server.handler.DefaultWebFilterChain.filter(DefaultWebFilterChain.java:119)
    |_                   Mono.defer ⇢ at org.springframework.web.server.handler.DefaultWebFilterChain.filter(DefaultWebFilterChain.java:119)
    |_            Mono.switchIfEmpty ⇢ at org.springframework.security.web.server.authorization.AuthorizationWebFilter.filter(AuthorizationWebFilter.java:46)
    |_                   checkpoint ⇢ org.springframework.security.web.server.authorization.AuthorizationWebFilter [DefaultWebFilterChain]
    |_                   Mono.defer ⇢ at org.springframework.web.server.handler.DefaultWebFilterChain.filter(DefaultWebFilterChain.java:119)
    |_ FluxMap$MapSubscriber.onComplete ⇢ at reactor.netty.channel.FluxReceive.terminateReceiver(FluxReceive.java:391)
```

Exceptions with debug mode enabled

```
Error has been observed at the following site(s):
    |_                Mono.doOnNext ⇢ at io.spring.workshop.tradingservice.TradingCompanyClient.getTradingCompany(TradingCompanyClient.java:38)
    |_            Mono.switchIfEmpty ⇢ at io.spring.workshop.tradingservice.TradingCompanyClient.getTradingCompany(TradingCompanyClient.java:39)
    |_                 Mono.zipWith ⇢ at io.spring.workshop.tradingservice.QuotesController.quotesDetails(QuotesController.java:39)
    |_            Mono.switchIfEmpty ⇢ at org.springframework.http.codec.EncoderHttpMessageWriter.write(EncoderHttpMessageWriter.java:119)
    |_                 Mono.flatMap ⇢ at org.springframework.http.codec.EncoderHttpMessageWriter.write(EncoderHttpMessageWriter.java:123)
    |_                   checkpoint ⇢ Handler io.spring.workshop.tradingservice.QuotesController#quotesDetails(String) [DispatcherHandler]
    |_                 Mono.flatMap ⇢ at org.springframework.web.reactive.DispatcherHandler.lambda$handleResult$5(DispatcherHandler.java:172)
    |_            Mono.onErrorResume ⇢ at org.springframework.web.reactive.DispatcherHandler.handleResult(DispatcherHandler.java:171)
    |_                 Mono.flatMap ⇢ at org.springframework.web.reactive.DispatcherHandler.handle(DispatcherHandler.java:147)
    |_                   Mono.defer ⇢ at org.springframework.web.server.handler.DefaultWebFilterChain.filter(DefaultWebFilterChain.java:119)
    |_                   Mono.defer ⇢ at org.springframework.web.server.handler.DefaultWebFilterChain.filter(DefaultWebFilterChain.java:119)
    |_            Mono.switchIfEmpty ⇢ at org.springframework.security.web.server.authorization.AuthorizationWebFilter.filter(AuthorizationWebFilter.java:46)
    |_                   checkpoint ⇢ org.springframework.security.web.server.authorization.AuthorizationWebFilter [DefaultWebFilterChain]
    |_                   Mono.defer ⇢ at org.springframework.web.server.handler.DefaultWebFilterChain.filter(DefaultWebFilterChain.java:119)
    |_ FluxMap$MapSubscriber.onComplete ⇢ at reactor.netty.channel.FluxReceive.terminateReceiver(FluxReceive.java:391)
```

Exceptions with debug mode enabled

```
Error has been observed at the following site(s):
    |_              Mono.doOnNext ⇢ at io.spring.workshop.tradingservice.TradingCompanyClient.getTradingCompany(TradingCompanyClient.java:38)
    |_          Mono.switchIfEmpty ⇢ at io.spring.workshop.tradingservice.TradingCompanyClient.getTradingCompany(TradingCompanyClient.java:39)
    |_              Mono.zipWith ⇢ at io.spring.workshop.tradingservice.QuotesController.quotesDetails(QuotesController.java:39)
    |_          Mono.switchIfEmpty ⇢ at org.springframework.http.codec.EncoderHttpMessageWriter.write(EncoderHttpMessageWriter.java:119)
    |_              Mono.flatMap ⇢ at org.springframework.http.codec.EncoderHttpMessageWriter.write(EncoderHttpMessageWriter.java:123)
    |_               checkpoint ⇢ Handler io.spring.workshop.tradingservice.QuotesController#quotesDetails(String) [DispatcherHandler]
    |_              Mono.flatMap ⇢ at org.springframework.web.reactive.DispatcherHandler.lambda$handleResult$5(DispatcherHandler.java:172)
    |_          Mono.onErrorResume ⇢ at org.springframework.web.reactive.DispatcherHandler.handleResult(DispatcherHandler.java:171)
    |_              Mono.flatMap ⇢ at org.springframework.web.reactive.DispatcherHandler.handle(DispatcherHandler.java:147)
    |_               Mono.defer ⇢ at org.springframework.web.server.handler.DefaultWebFilterChain.filter(DefaultWebFilterChain.java:119)
    |_               Mono.defer ⇢ at org.springframework.web.server.handler.DefaultWebFilterChain.filter(DefaultWebFilterChain.java:119)
    |_          Mono.switchIfEmpty ⇢ at org.springframework.security.web.server.authorization.AuthorizationWebFilter.filter(AuthorizationWebFilter.java:46)
    |_               checkpoint ⇢ org.springframework.security.web.server.authorization.AuthorizationWebFilter [DefaultWebFilterChain]
    |_               Mono.defer ⇢ at org.springframework.web.server.handler.DefaultWebFilterChain.filter(DefaultWebFilterChain.java:119)
    |_ FluxMap$MapSubscriber.onComplete ⇢ at reactor.netty.channel.FluxReceive.terminateReceiver(FluxReceive.java:391)
```

Reactive Developer productivity

Soon coming as a jvm agent

Mobile Backend
Scale even with different traffic latency
Connected experience (notifications...)
Volume of clients

Soon coming as a jvm agent

Speaking of which...

Mobile Backend
Scale even with different traffic latency
Connected experience (notifications...)
Volume of clients

BlockHound !

https://github.com/reactor/BlockHound

Mobile Backend
Scale even with different traffic latency
Connected experience (notifications...)
Volume of clients

Detecting blocking code in reactive stack

```
BlockHound.install();

someMono.doOnSuccess(data -> {

     TradingCompany tradingCompany = this.restTemplate.exchange(
         get("http://localhost:8082/details/{ticker}", ticker),
         TradingCompany.class)
         .getBody();

     return tradingCompany;

});
```

Detecting blocking code in reactive stack

```
java.lang.Error: Blocking call! java.net.Socket#connect
  at reactor.blockhound.BlockHound$Builder.lambda$new$0(BlockHound.java:154)
~[blockhound-1.0.0.jar:na]
  at reactor.blockhound.BlockHound$Builder.lambda$install$8(BlockHound.java:254)
~[blockhound-1.0.0.jar:na]
  at reactor.blockhound.BlockHoundRuntime.checkBlocking(BlockHoundRuntime.java:43)
~[blockhound-1.0.0.jar:na]
  at java.net.Socket.connect(Socket.java) ~[na:1.8.0_144]
  at sun.net.NetworkClient.doConnect(NetworkClient.java:180) ~[na:1.8.0_144]
  at sun.net.www.http.HttpClient.openServer(HttpClient.java:463) ~[na:1.8.0_144]
  at sun.net.www.http.HttpClient.openServer(HttpClient.java:558) ~[na:1.8.0_144]
  at sun.net.www.http.HttpClient.<init>(HttpClient.java:242) ~[na:1.8.0_144]
  at sun.net.www.http.HttpClient.New(HttpClient.java:339) ~[na:1.8.0_144]
  at sun.net.www.http.HttpClient.New(HttpClient.java:357) ~[na:1.8.0_144]
```

Detecting blocking code in reactive stack

```
java.lang.Error: Blocking call! java.net.Socket#connect
  at reactor.blockhound.BlockHound$Builder.lambda$new$0(BlockHound.java:154)
~[blockhound-1.0.0.jar:na]
  at reactor.blockhound.BlockHound$Builder.lambda$install$8(BlockHound.java:254)
~[blockhound-1.0.0.jar:na]
  at reactor.blockhound.BlockHoundRuntime.checkBlocking(BlockHoundRuntime.java:43)
~[blockhound-1.0.0.jar:na]
  at java.net.Socket.connect(Socket.java) ~[na:1.8.0_144]
  at sun.net.NetworkClient.doConnect(NetworkClient.java:) ~[na:1.8.0_144]
  at sun.net.www.http.HttpClient.openServer(HttpClient.java:463) ~[na:1.8.0_144]
  at sun.net.www.http.HttpClient.openServer(HttpClient.java:558) ~[na:1.8.0_144]
  at sun.net.www.http.HttpClient.<init>(HttpClient.java:242) ~[na:1.8.0_144]
  at sun.net.www.http.HttpClient.New(HttpClient.java:339) ~[na:1.8.0_144]
  at sun.net.www.http.HttpClient.New(HttpClient.java:357) ~[na:1.8.0_144]
```

Quickly detect during pre-production workflows that will likely degrade your reactive stack experience

More reactive microservices ?

80% of apps generated by start.spring.io are using a SQL driver

80% of apps generated by start.spring.io are using a SQL driver

r2dbc.io

What should I look after I successfully have written my first Reactive apps ?

Monitoring with Micrometer

micrometer.io

Networking with RSocket

rsocket.io

Having fun developing with Kotlin

kotlinlang.org

So In Conclusion

What are you priorities ?

Time To Market / Dev Productivity

Day 2 Operations / Observability

[...]

Running costs / efficiency

What are you priorities ?

Time To Market / Dev Productivity

With Reactive

You can actually be more productive in writing state of the art, scalable microservices

Day 2 Operations / Observability

[...]

Running costs / efficiency

What are you priorities ?

Time To Market / Dev Productivity

Day 2 Operations / Observability

[...]

Running costs / efficiency

What are you priorities ?

Time To Market / Dev Productivity

With Reactive
Your Applications are designed to be resilient under stress and naturally lower maintenance

Day 2 Operations / Observability

[...]

Running costs / efficiency

What are you priorities ?

Time To Market / Dev Productivity

Day 2 Operations / Observability

[...]

Running costs / efficiency

What are you priorities ?

Time To Market / Dev Productivity

With Reactive
You run more for less

Day 2 Operations / Observability

[...]

Running costs / efficiency

What are you priorities ?

Time To Market / Dev Productivity

Day 2 Operations / Observability

[...]

Running costs / efficiency

Establish the right project to experiment
Reactive Programming with

Do it progressively, migrate a few dedicated backend routes then an entire application

You can already parallelize some workloads even on your blocking stack

Qualify a few benefits identified in this session

Establish your own benchmark to verify the outcome !

Use all the developers tools you can to accelerate your reactive adoption and build trust

Once the learning curve price has been paid and the benefits realized- it’s usually difficult to come back to blocking solutions.

In fact it’s difficult in today’s world of distributed systems to not find a reason for giving a go to any Reactive stack.

In fact it’s difficult in today’s world of distributed systems to not find a reason for giving a go to any Reactive stack.

It’s just designed for them

THANK YOU


@smaldini


smaldini