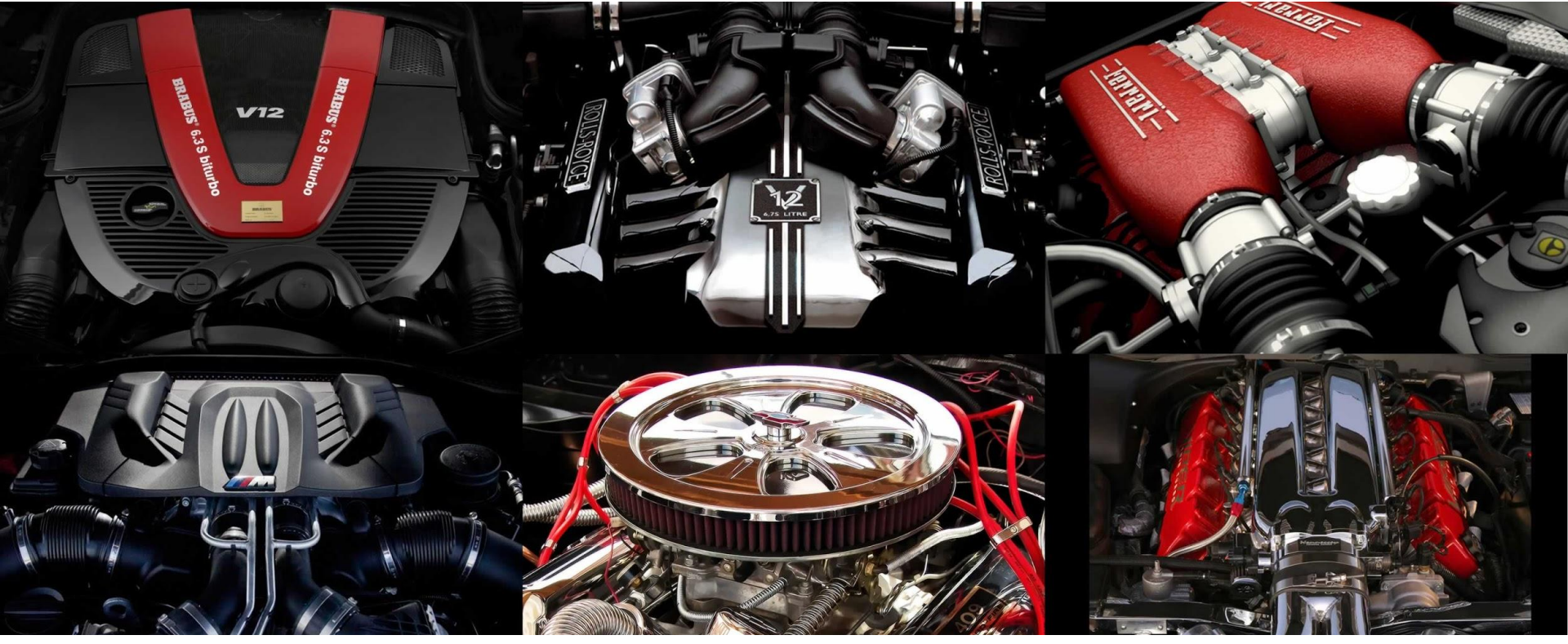
Using Java SE 8 with Java EE 7 in the Real World

Michael Nascimento Santos - @mr_ _m
misterm@gmail.com

Agenda

- Where to run?
- Lambda
- parallelStream() vs Concurrency Utilities
- java.time with JSF / JPA / JAX-RS
- Parameter retention
- Default and static methods in interfaces

Where to deploy?



We use (most of the time) open-source application servers

We conducted a study of where to deploy our applications based on compatibility and performance

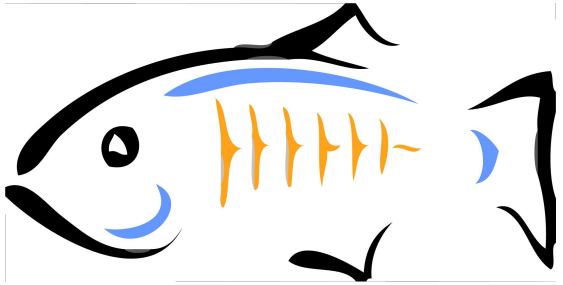
Where to run?

- Test suite
  - JSF + CDI (ViewAccessScoped, @PostConstruct, Stateless and Stateful, using JDBC, using JPA…)
  - JAX-RS + CDI
  - JAX-WS + CDI
  - @Async / @Schedule
  - Still growing! :-)

Servers

9.2.6

8.2



GlassFish 4.1

Where to run?

- TomEE: had no release at the time that supported Java EE 7

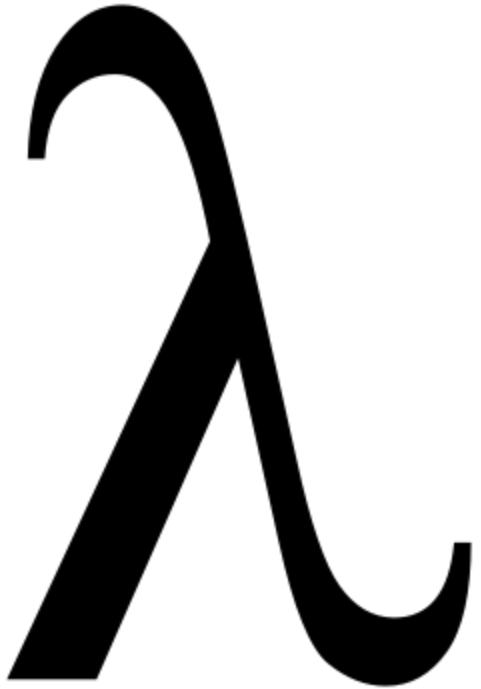
- Jetty:
  - EJBs require OpenEJB
  - That requires OpenWebBeans
  - That requires the rest of the world
  - And lot of errors :-S

Where to run?

- GlassFish and WildFly
  - Little glitches at first
    - After migrating more projects, now we have found more, all related to EE not SE though
  - Very close performance characteristics, with slight advantage for GlassFish
  - WildFly selected though

Write tests that cover your usage patterns!

Lambda



Lambda

- Doesn’t replace JPQL / SQL / Whatever QL

- May be suitable when:
  - Several operations processing the same data set
  - Complex business logic
  - Elements are naturally available in RAM

parallelStream()
vs
Concurrency

parallelStream() vs Concurrency

- parallelStream() was not conceived having Java EE in mind :-S
  - No process isolation
  - No pool configuration
- Use stream() as is / Concurrency Utilities / write a replacement (let me know!)

java.time with ...

Parameter retention & multiple annotations

Default and static in interfaces

Default and static methods in interfaces

- Default
  - Interface evolution
  - Possible to implement without state
  - Flexible class hierarchy
  - Enabling lambda
- Static
  - Factories
- In Java 9, private methods!

Summary

- Some stuff brings immediate value (lambda & streams, default & static methods in interfaces)
- Others require effort (java.time + Java EE, parameter retention)
- Others... better not! (parallelStream())
- Java EE 8 will improve things

Thanks to...

- Janario Oliveira
- João Bosco Monteiro
- Rodrigo Santos

Thank you!

misterm@gmail.com
@mr _ _ m

References

- A Java™ Parallel Calamity (a little bit exaggerated)
- Think twice before using Java 8 parallel streams
- Custom thread pool in Java 8 parallel stream
- When to use parallel streams
- JPA Converters (impl, not tested!)

References

- More JPA converters (untested)
- Issue for java.time support in Hibernate (5.0)
- Issue for java.time support in JPA
- Issue for Java SE 8 CDI support
- Repeating annotations in Java EE