

# Serverless Architectures with Java

...

Bill Koch

About me...



I do not receive kickbacks from Pivotal/VMWare



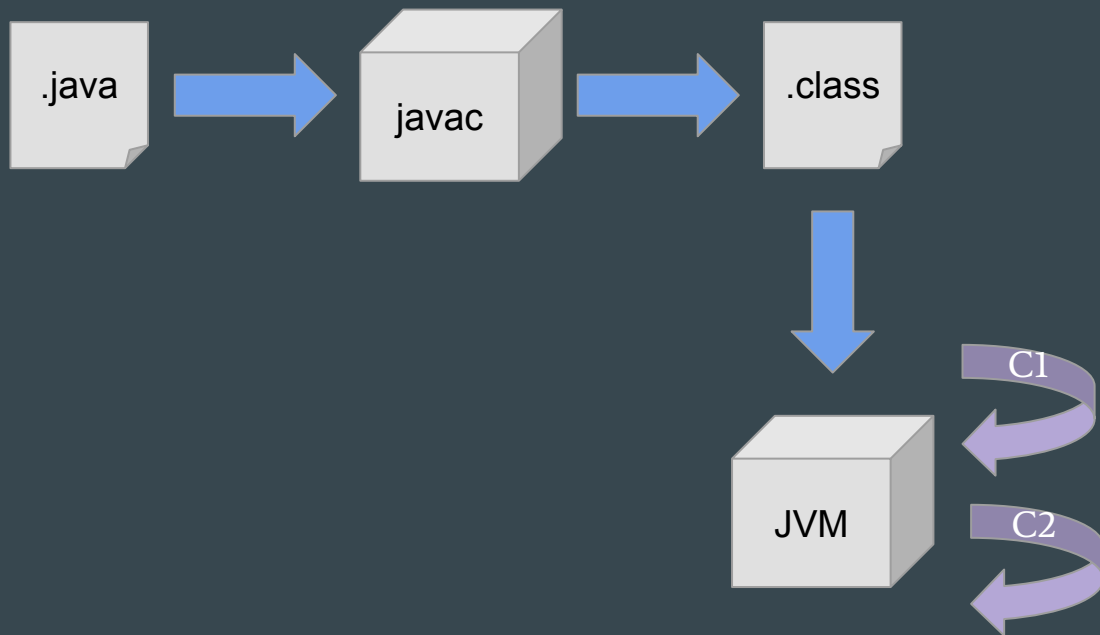
Is Java + Spring Poorly Suited for Serverless?

It depends.

# Serverless Considerations

- Application startup time
- Memory consumption
- CPU consumption

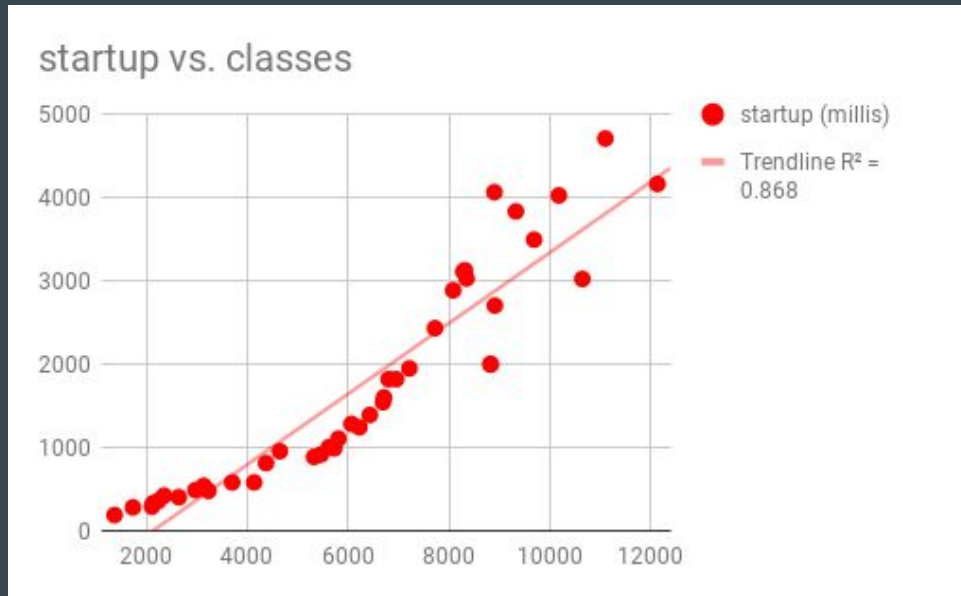
# Why is Java “slow” to start up?



# Why is Spring “slow” to start up?

## Reflection

- @Autowired, @Controller, ...
- Auto configuration
- <https://spring.io/blog/2018/12/12/how-fast-is-spring>



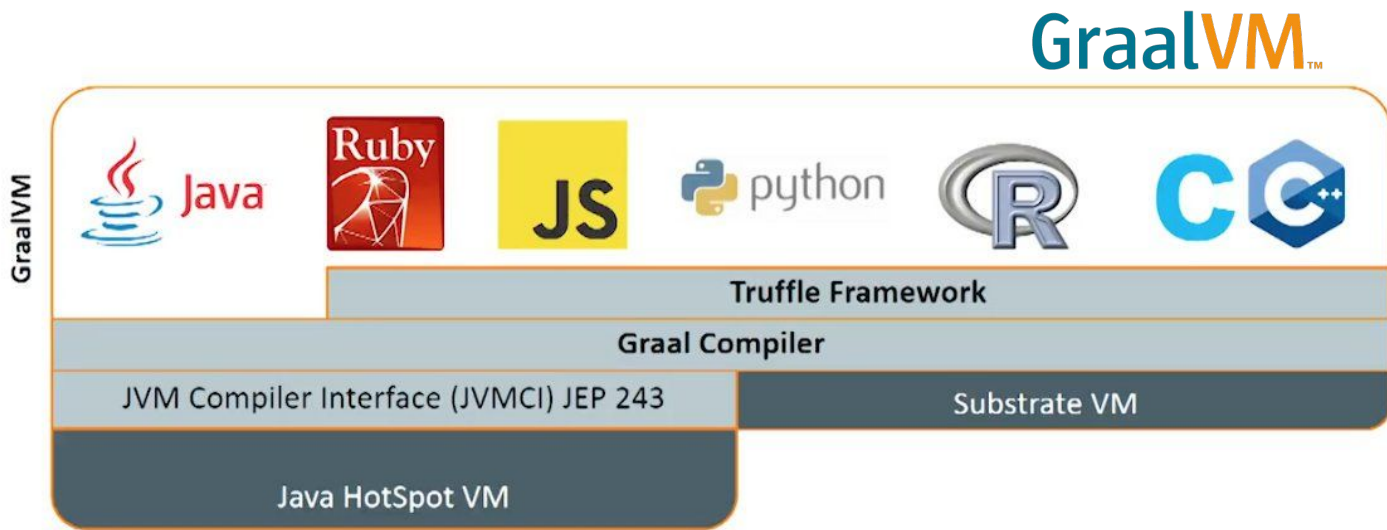
# What is Graal?

# Graal (the JIT Compiler)

- JVMCI (JEP 243)
- Written in Java
- Baked into OpenJDK 9+
- Can be enabled as a C2 JIT replacement
- Twitter in production <https://youtu.be/G-vlQaPMAxg>
  - ~12% CPU reduction with no additional tuning
  - ~3% drop in GC time



# Graal (the VM)



# Graal (native-image)

- GraalVM add-on to generate machine code via AOT compilation
- SubstrateVM
- libgraal

# Micronaut

- Codes like Spring Boot without reflection\*
  - Annotation processors triggered at compile time
    - Allows compilers to perform better inlining and optimizations
    - Reduced memory consumption
- Developed by ex-Groovy/Grails engineers from Pivotal
- Rich module ecosystem
  - Netflix OSS
  - Zipkin
  - Micrometer
  - ...

# Rough edges with Graal native-image

A lot of Java frameworks leverage reflection...

- Manual configuration of reflection use
- @AutomaticFeature



# What about Spring + Graal native-image?

- <https://github.com/spring-projects/spring-framework/wiki/GraalVM-native-image-support>
- Initial support built into Spring 5.1 (spring-boot 2.1.x)
- Spring 5.2 (spring-boot 2.2.x) dedicated to finding issues with Graal GA
- Spring 5.3 will automatically provide reflection configuration to Graal native-image (@AutomaticFeature)

# Credits

- <https://spring.io/blog/2018/12/12/how-fast-is-spring>
- <https://www.techtalks.lk/blog/graalvm-holy-grail-for-interoperability>
- [https://en.wikipedia.org/wiki/File:Poison\\_Help.svg](https://en.wikipedia.org/wiki/File:Poison_Help.svg)