

GIAC

全球互联网架构大会

GLOBAL INTERNET ARCHITECTURE CONFERENCE

如何构建云原生Java应用

姜宁 华为 开源能力中心技术专家



TOP100Summit

全球软件案例研究峰会

时间：11月15~17日

地点：北京国际会议中心

100个年度最值得学习案例

MPD工作坊（深圳站）

时间：9月21~22日

地点：深圳博林圣海伦酒店

20个3小时大时段沙盘课程

Make Professional
mpd

100

Make Professional
mpdMake Professional
mpd**MPD工作坊（北京站）**

时间：7月06~07日

地点：北京国家会议中心

20个3小时大时段沙盘课程

MPD工作坊（上海站）

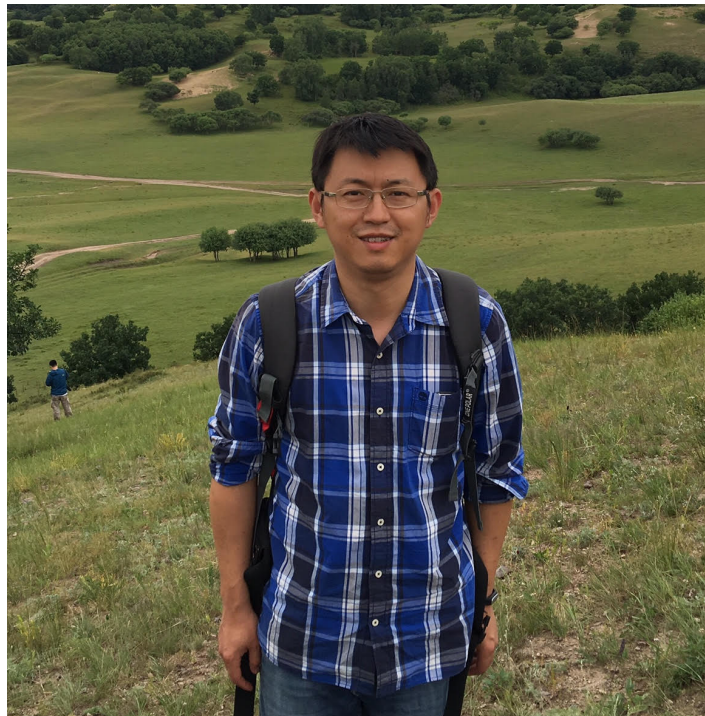
时间：10月26~27日

地点：上海

20个3小时大时段沙盘课程

关于我

- 目前就职于华为开源能力中心
- 前红帽软件主任软件工程师
- Apache软件基金会会员
- 参与Apache软件基金会多个Java 中间件项目



云原生是什么？

- Cloud Native 定义
 - 起源于Pivotal
 - 适应云计算的应用构建方式。
- CNCF 云原生计算基金会的最初定义
 - 应用容器化
 - 面向微服务架构
 - 应用支持容器的编排调度



云原生应用的特点

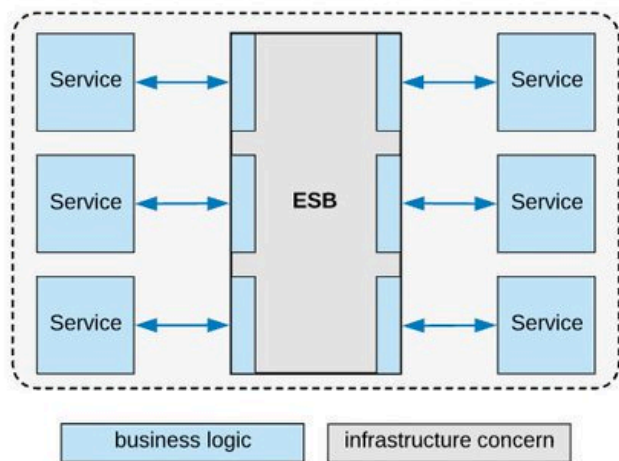
- 符合12因素应用
 - 使用**标准化**流程自动配置
 - 与操作系统尽可能**划清界限**
 - 适合**部署**在现代的**云计算平台**
 - 将开发环境和生产环境的**差异降至最低**
- 面向微服务架构
- 基于API的协作
- 抗脆弱性



服务化到云原生架构演进

Service Oriented Architecture

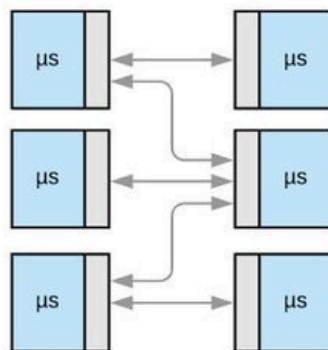
(Smart pipes, dumb endpoints)



聪明管道 配 老旧节点

Microservices Architecture

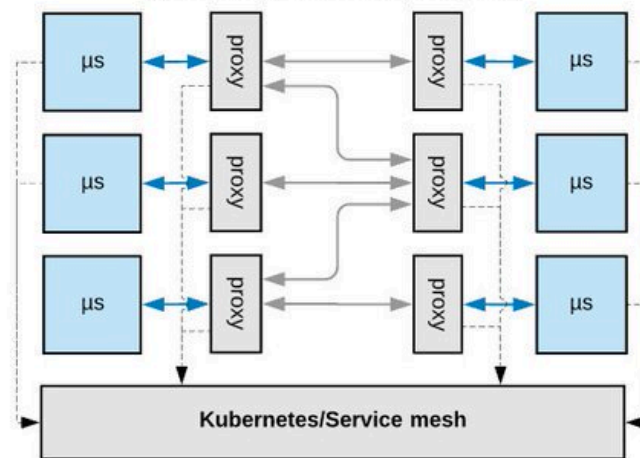
(Smart endpoints, dumb pipes)



聪明节点 配 老旧管道

Cloud Native Architecture

(Infrastructure focused smart platform, business logic focused smart services)

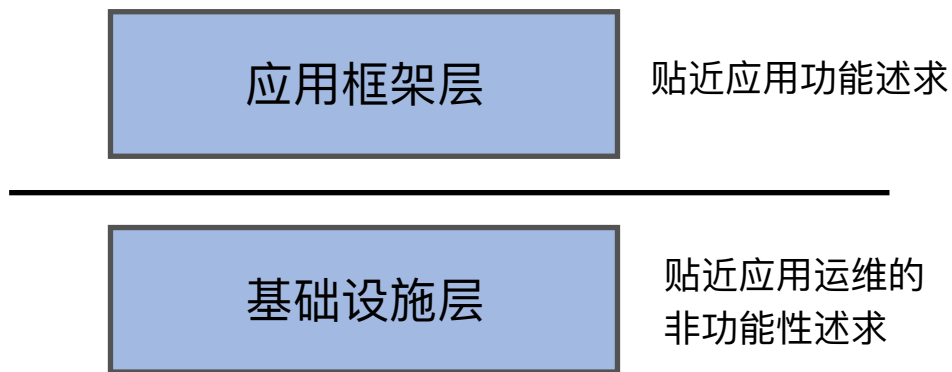


聪明平台 配 聪明的服务



云原生技术栈

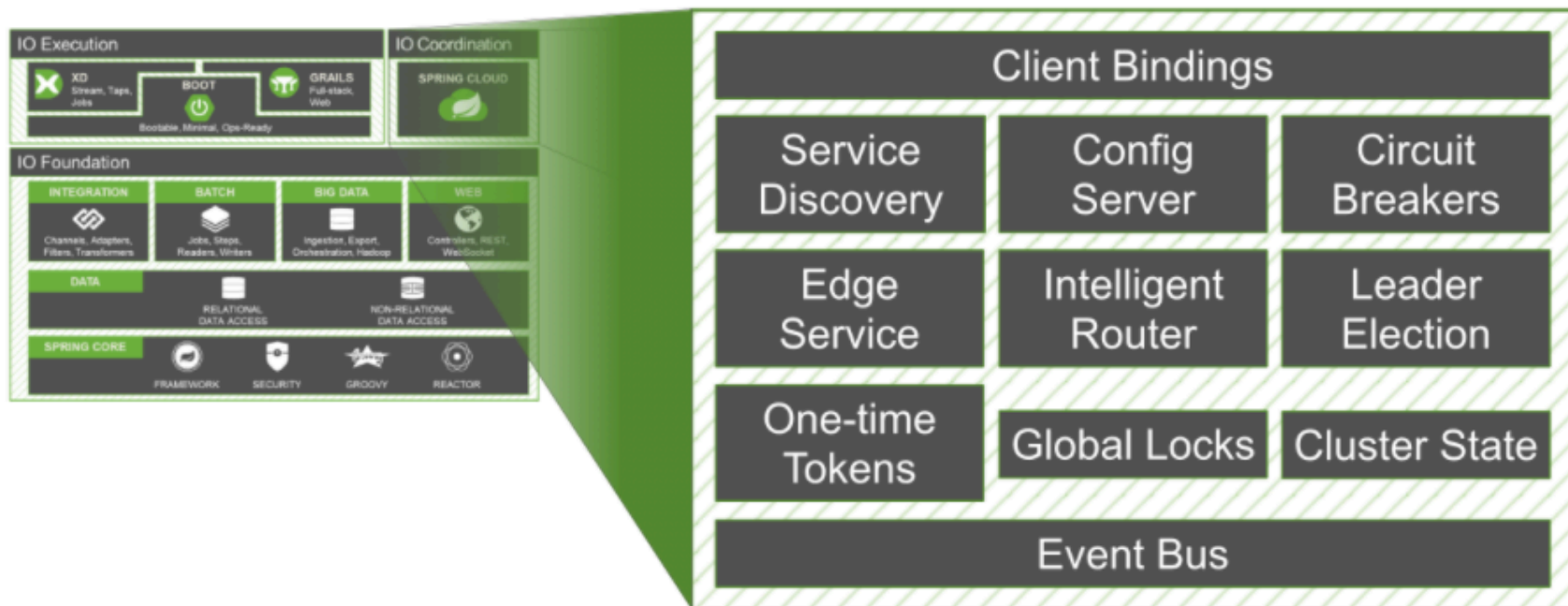
- 容器技术
- 服务网格
- 微服务
- 不可变基础设施
- 声明式API



云原生应用对容器化以及计算设施友好



云原生Java应用的现状



这些功能都需要应用框架来实现吗？
应用是容器友好的吗？

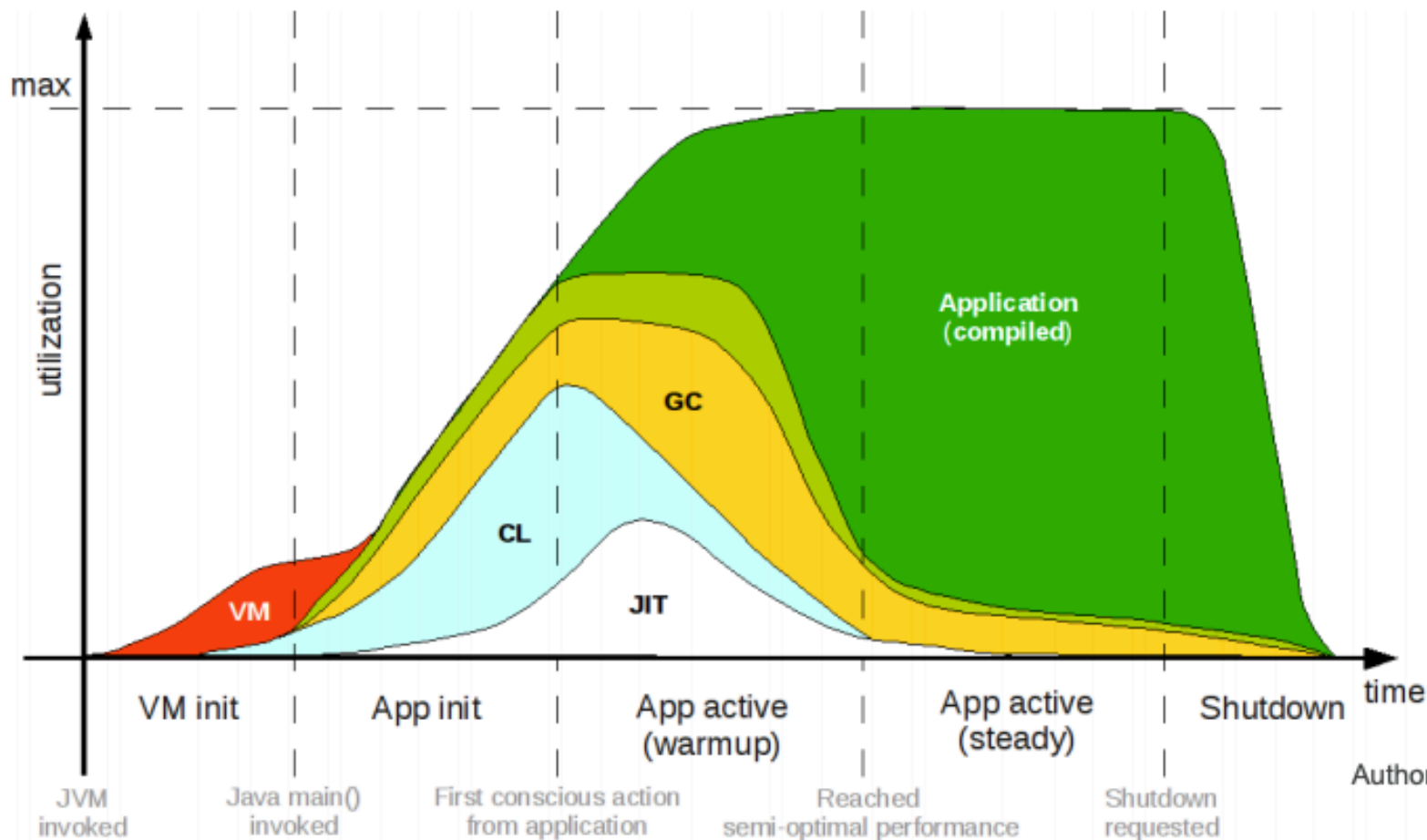


传统Java应用的问题?

- 轻量容器的产生, 让JVM的优势变成劣势
 - 依赖JVM — 容器镜像大 (200MB)
 - 依赖JVM — 内存占用大 (200MB)
 - 应用代码 — Spring Boot (30~100MB)
- 微服务架构引入
 - 服务颗粒度变小, 部署频繁
 - JVM自身启动加载耗时是一个问题



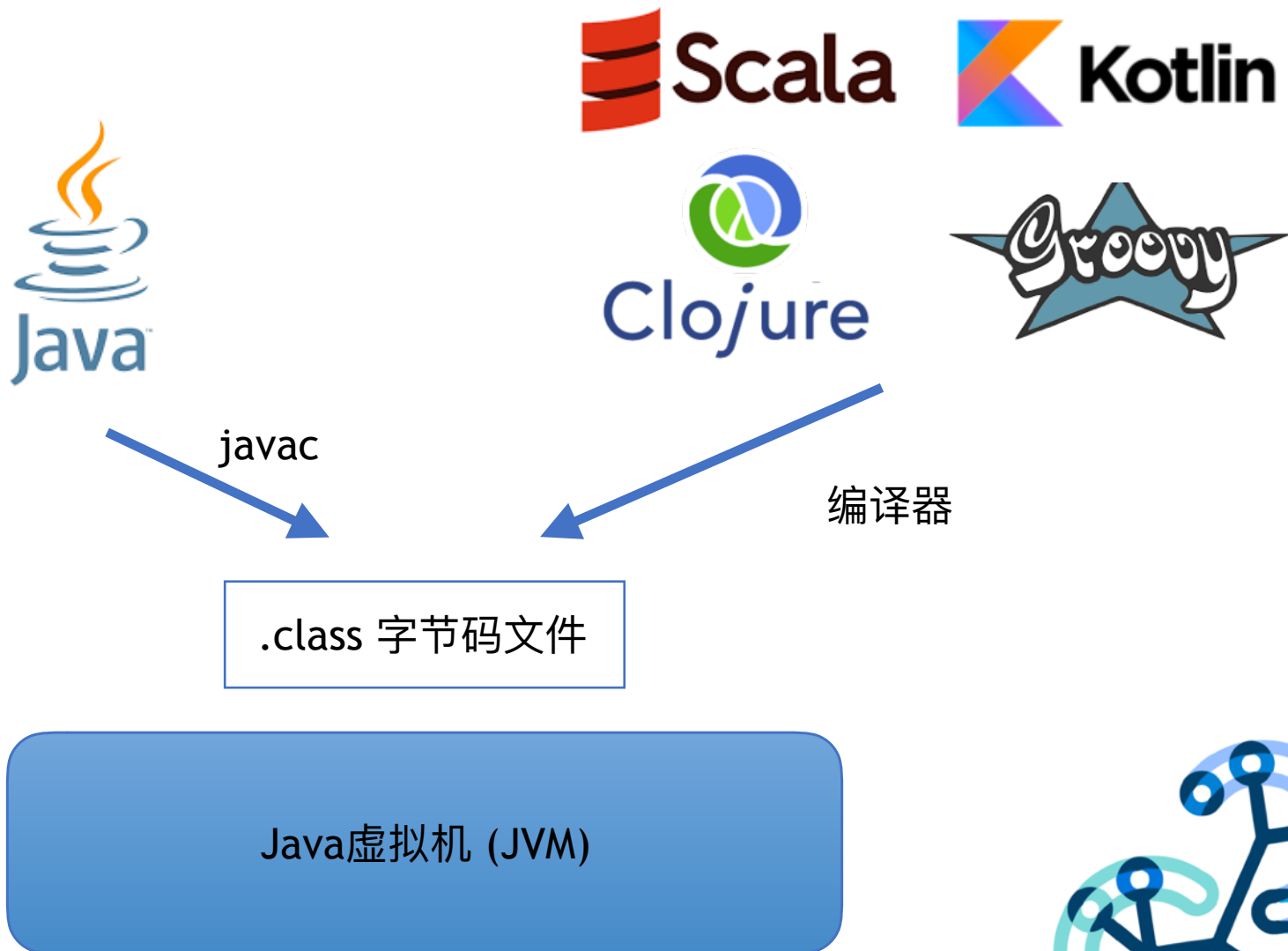
Java应用的各模块执行时间分布



Author: Aleksey Shipilev



Java代码执行过程



Java 代码执行过程

.class 字节码

解释器

- 执行缓慢
- 可以立即生效
- 可以收集程序运行信息

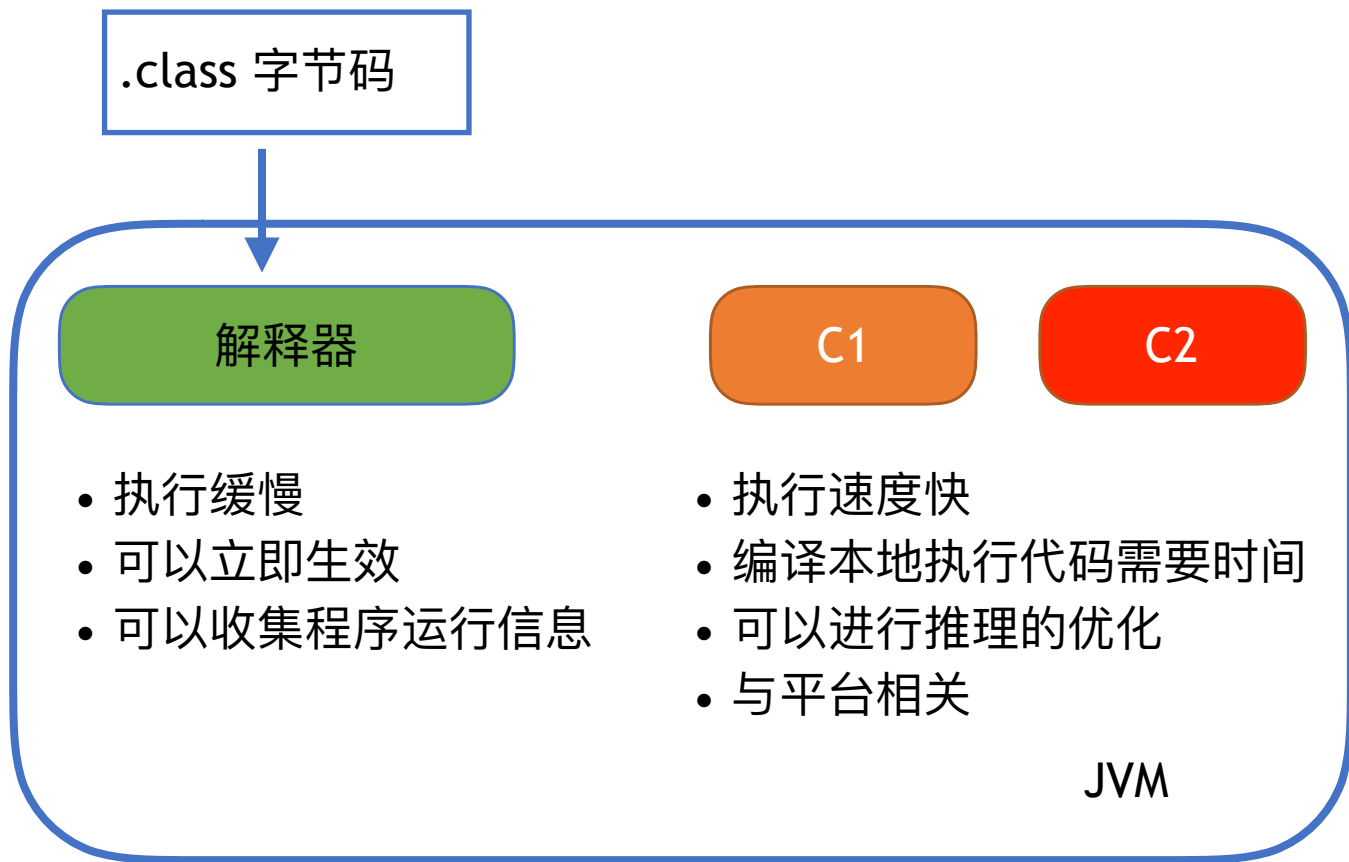
即时编译器 (JIT)

- 执行速度快
- 编译本地执行代码需要时间
- 可以进行推理的优化
- 与平台相关

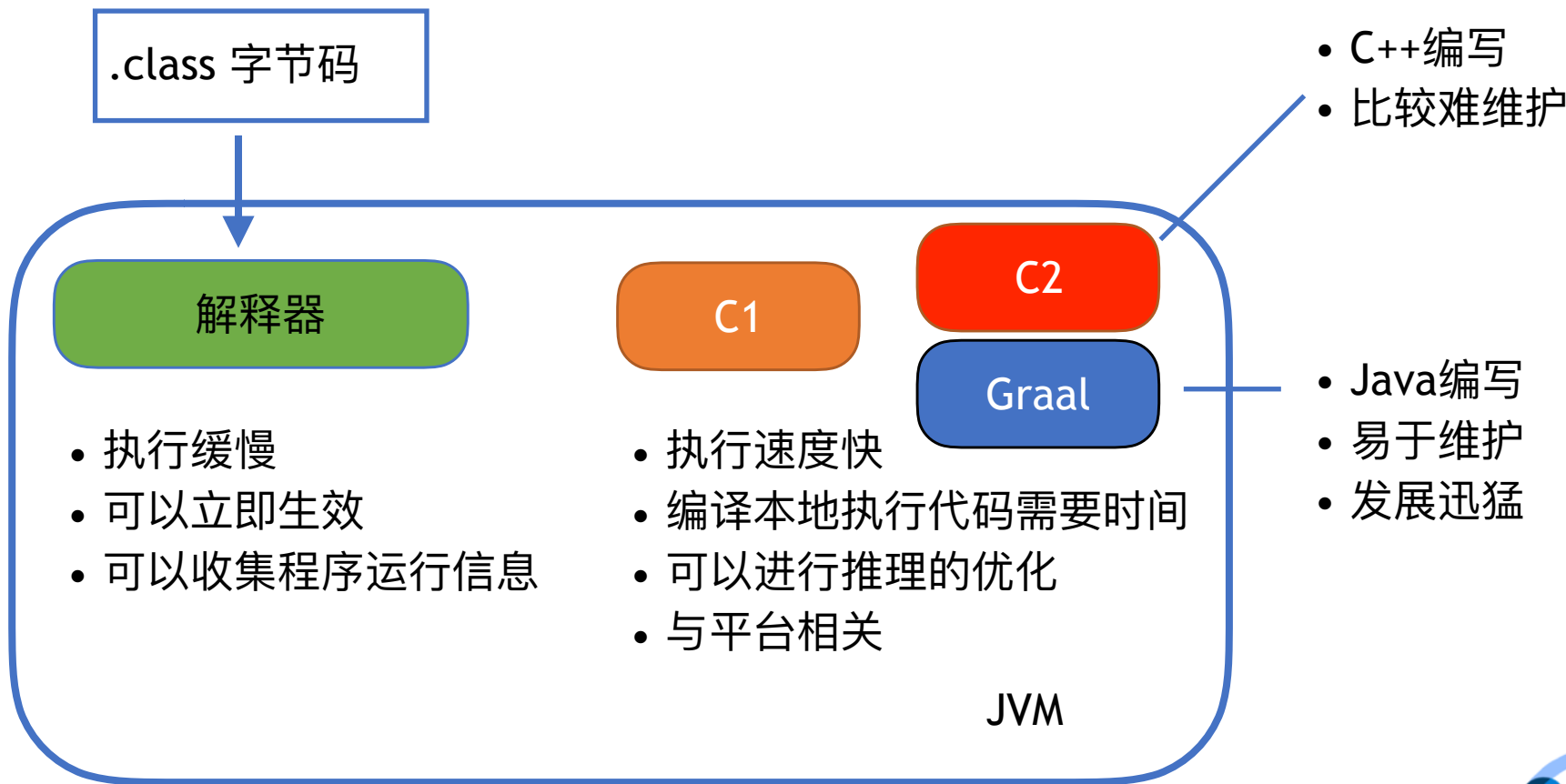
JVM



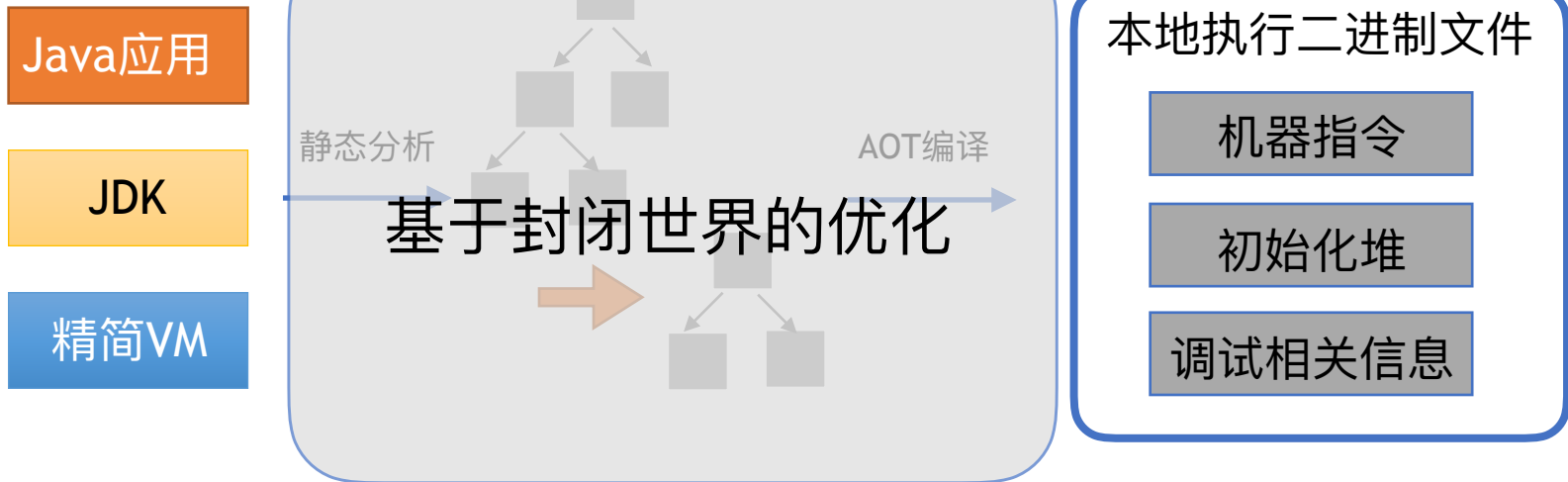
Java 代码执行过程



Java 代码执行过程



使用Graal做静态编译



从应用, JDK, 以及精简VM中获取所有的Java class信息

获取方法, 成员变量以及类实例

经过静态编译的应用可以不依赖于JDK运行, 同时也没有类加载



“Hello World”静态编译的效果

C

启动时间: < 10ms

内存使用: 450KB

文件大小: 100KB

Java/JVM

启动时间: 40 ms

内存使用: 24 MB

文件大小: 140M

Java/SVM

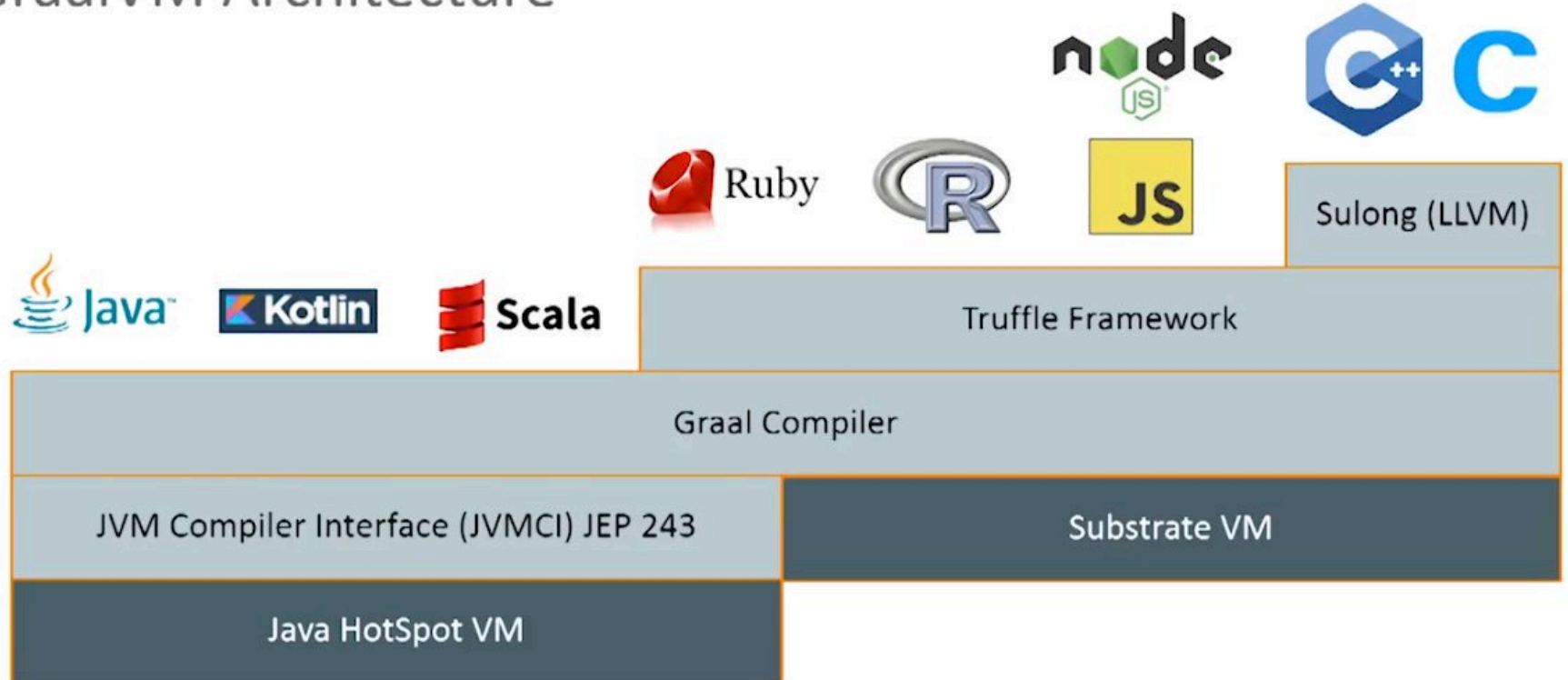
启动时间: < 10ms

内存使用: 850KB

文件大小: 220KB



GraalVM Architecture

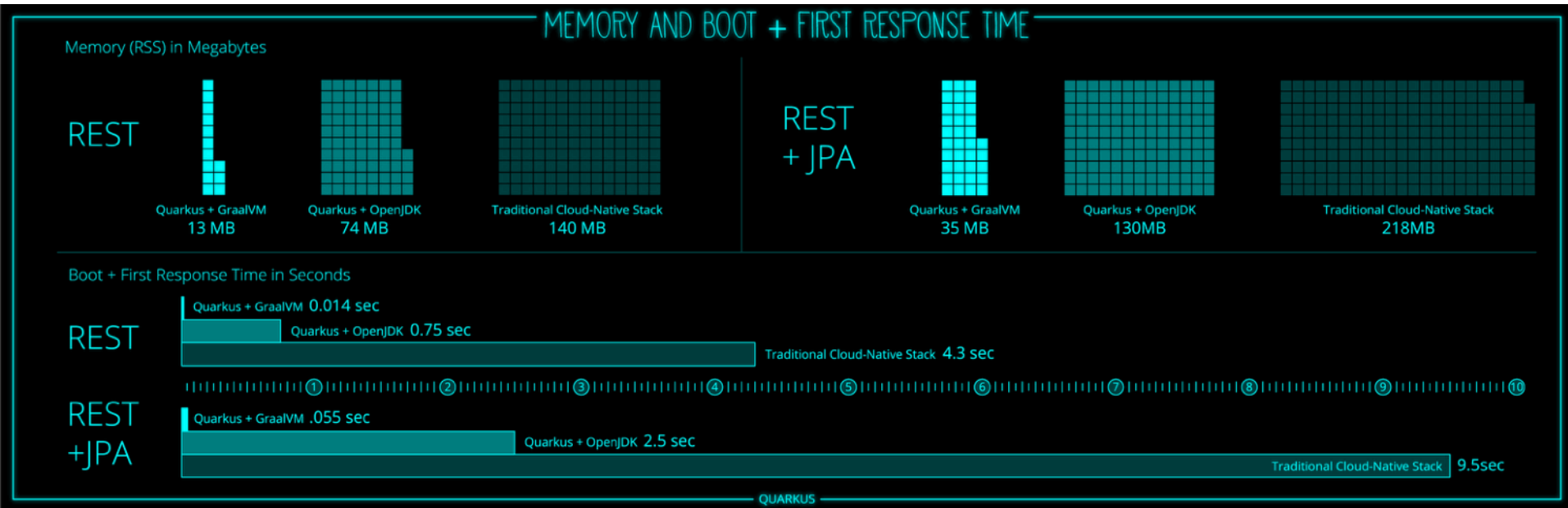


Quarkus

- 夸克思, Supersonic Subatomic Java
- Quarkus is a Kubernetes Native Java framework tailored for GraalVM and HotSpot, crafted from best-of-breed Java libraries and standards.
- Quarkus 是一个针对K8s原生Java框架, 这个框架针对GraalVM和HotSpot进行量身定制, 并且针对很多优秀的Java库以及标准进行优化。



Quarkus基于容器的应用



应用启动速度快了百倍， Image大小变成了原来的十分之一



Quarkus支持的框架

VERT.X



REST
Eas



Infinispan



因为要支持静态编译，需要避免使用动态加载类方法，需要对这些库进行修改



Quarkus支持的扩展

- Core
- Web
- Data
- Messaging
- Reactive
- Cloud
- Observability
- Security
- Integration
- Serialization
- Miscellaneous
- Migration
- Alternative Languages



Quarkus环境准备

- 选择一个适合的IDE
- 安装JDK 1.8+, 设置 JAVA_HOME
- 安装GraalVM, 设置GRAALVM_HOME
- 配置C语言开发环境
 - Linux 安装GCC
 - macOS 执行 `xcode-select --install`
- 安装Docker
 - 可以编译docker native image



Quarkus入门指南

- Hello World 开发
 - GraalVM 安装
 - 设置Maven, Gradle 插件
 - 生成工程代码
 - 配置扩展
 - 编译调试
 - 本地执行代码生成



Quarkus 从代码到镜像

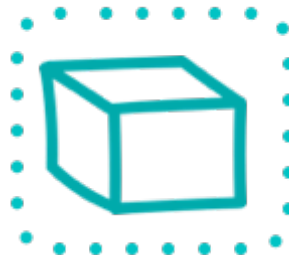
编写应用代码



测试运行



生成容器镜像



生成本地
执行代码

容器化



Quarkus 开发与测试

- 典型的Web应用开发
- 如何使用WebSocket
- 添加扩展模块
- 快速获取反馈的dev 模式
- 测试的应用



<https://github.com/WillemJiang/smart-park-demo>



Quarkus相关文档教程

- Quarkus Github地址

<https://github.com/quarkusio>

- Quarkus 快速入门

<https://quarkus.io/get-started/>

- Quarkus 用户手册

<https://quarkus.io/guides/>

- Quarkus 实例代码

<https://github.com/quarkusio/quarkus-quickstarts>



小结

- 云原生应用的特点
- 传统Java应用的问题
- 借助GraalVM提升Java应用
- 通过Quarkus项目构建新一代的Java云原生应用





欢迎关注msup微信公众账号

关注大会微信公共账号，及时了解大会动态、
日程及每日更新的案例！

