Tristan Mahinay

- **Senior Technical Specialist and Java SME** - IBM CIC Philippines
- **Java User Group Leader** - Java User Group Philippines
- **Open Source Contributor** - Quarkus
- **Blog Author**- Foojay

@ph_tantan

rjtmahinay

me@rjtmahinay.com

Tristan Mahinay

linkedin.com/rjtmahinay

The Value of Quarkus

A developer's perspective

Let’s explore Quarkus

What is Quarkus?

Supersonic. Subatomic. Java

Kube-native

- Single-step Deployment
- Application Configuration
- Health & Metrics
- Tracing and Debugging
- Remote Development

Cloud-native

- Kubernetes
- Fast start-up time
- Reduced reflection usage
- GraalVM Native Support

Imperative and Reactive Code

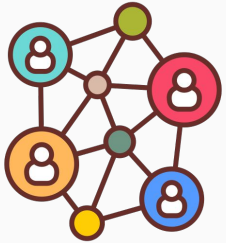
- Microservices (REST/SOAP)
- Reactive Programming
- Function-as-a-Service (FaaS) and Serverless
- Event-driven Architecture

What is Quarkus?

Supersonic. Subatomic. Java

Developer-centric

- Live reload/coding
- Remote Development
- Dev UI/Services
- Continuous Testing
- Extensions



Standards

- Vert.x
- Eclipse Microprofile
- Micrometer
- Jakarte EE

DEV UI Interface

Dev UI

Extensions

quarkus-java 1.0-SNAPSHOT

Back to the previous Dev UI

- Extensions
- Configuration
- Continuous Testing
- Dev Services
- Build Metrics
- Information

ArC
Build time CDI dependency injection
Beans 96
Observers 4
Removed components 104

Cache
Enable application data caching in CDI beans
Caches

Datasources
Configure your datasources
Reset

Hibernate ORM
Define your persistent model with Hibernate ORM and Jakarta Persistence
Persistence Units
Entity Types
Named Queries

Kubernetes
Generate Kubernetes resources from annotations
Kubernetes Manifests

Micrometer metrics
Instrument the runtime and your application with dimensional metrics using Micrometer.
Prometheus

OpenShift
Generate OpenShift resources from annotations
Deploy to OpenShift

SmallRye Health
Monitor service health
Health
Health UI

SmallRye OpenAPI
Document your REST APIs with OpenAPI - comes with Swagger UI
Schema yaml
Schema json
Swagger UI

Caffeine
Caffeine is a high performance, near optimal caching library.
Version 3.1.5

Container Image
Build container images of your

Agroal - Database connection pool
Pool JDBC database connections

Eclipse Vert.x
Write reactive applications with the

Hibernate ORM with Panache
Simplify your persistence code for

JDBC Driver - H2
Connect to the H2 database via JDBC

Quarkus 3.1.2.Final

Server Testing

```
--\___\_\____/_/ |_/_/|_/_/|_|\____/___/
2023-06-29 00:08:06,591 WARN [] (Quarkus Main Thread) Unrecognized configuration key "quarkus.employee.job.title" was provided; it will be ignored; verify that the dependency extension for this configuration is set or that you did not make a typo
2023-06-29 00:08:06,591 WARN [] (Quarkus Main Thread) Unrecognized configuration key "quarkus.employee.name" was provided; it will be ignored; verify that the dependency extension for this configuration is set or that you did not make a typo
2023-06-29 00:08:07,864 INFO [io.qua.boo.run.Timing] (Quarkus Main Thread) quarkus-java 1.0-SNAPSHOT on JVM (powered by Quarkus 3.1.2.Final) started in 9.126s. Listening on: http://localhost:8080
2023-06-29 00:08:07,867 INFO [io.qua.boo.run.Timing] (Quarkus Main Thread) Profile dev activated. Live Coding activated.
2023-06-29 00:08:07,869 INFO [io.qua.boo.run.Timing] (Quarkus Main Thread) Installed features: [agroal, cache, cdi, config-yaml, hibernate-orm, hibernate-orm-panache, jdbc-h2, jdbc-mysql, kubernetes, micrometer, narayana-jta, rest-client
```

Extensions

Quarkus Platform and Quarkiverse

https://github.com/quarkiverse

https://github.com/quarkusio/quarkus-platform

QUARKUS

ABOUT LEARN EXTENSIONS COMMUNITY START CODING

Extensions

A Quarkiverse of extensions enhance your application just as project dependencies do.

Find an extension

Showing 616 extensions

Status

All

Category

- Integration
- Data
- Web
- Cloud
- Miscellaneous
- Security
- Reactive
- Serialization
- Messaging
- Observability

Agroal - Database connection pool

Pool JDBC database connections (included in Hibernate ORM)

Category: Data
Latest Version: 3.2.2.Final
Publish Date: Jul 21, 2023

Amazon Alexa

Write Amazon Alexa Skills with AWS Lambda or HTTPS endpoints

Category: Cloud
Latest Version: 2.0.0
Publish Date: Apr 12, 2023

Amazon Cognito User Pools

Connect to Amazon Cognito User Pools service

Category: Data
Latest Version: 2.4.1
Publish Date: Jul 15, 2023

Amazon DynamoDB

Connect to Amazon DynamoDB datastore

Category: Data
Latest Version: 2.4.1
Publish Date: Jul 15, 2023

Quarkus RedHat Build

https://code.quarkus.redhat.com/

QUARKUS 2.13 com.redhat.quarkus.platform

Red Hat < Back to code.quarkus.io

CONFIGURE YOUR APPLICATION

Group	org.acme	Version	1.0.0-SNAPSHOT
Artifact	code-with-quarkus	Java Version	17
Build Tool	Maven	Starter Code	Yes

CLOSE

0 Generate your application (alt + ⏎)

Filters origin:platform

Web

- RESTEasy Reactive [quarkus-resteasy-reactive] STARTER-CODE SUPPORTED
  A JAX-RS implementation utilizing build time processing and Vert.x. This extension is not compatible with the quarkus-resteasy extension, or any of the extensions that depend on it.
- RESTEasy Reactive Jackson [quarkus-resteasy-reactive-jackson] SUPPORTED
  Jackson serialization support for RESTEasy Reactive. This extension is not compatible with the quarkus-resteasy extension, or any of the extensions that depend on it
- RESTEasy Reactive JSON-B [quarkus-resteasy-reactive-jsonb] SUPPORTED
  JSON-B serialization support for RESTEasy Reactive. This extension is not compatible with the quarkus-resteasy extension, or any of the extensions that depend on it.
- RESTEasy Reactive JAXB [quarkus-resteasy-reactive-jaxb] TECH-PREVIEW
  JAXB serialization support for RESTEasy Reactive. This extension is not compatible with the quarkus-resteasy extension, or any of the extensions that depend on it.
- RESTEasy Reactive Kotlin Serialization [quarkus-resteasy-reactive-kotlin-serialization] STARTER-CODE
  Kotlin Serialization support for RESTEasy Reactive. This extension is not compatible with the quarkus-resteasy extension, or any of the extensions that depend on it.
- RESTEasy Reactive Qute [quarkus-resteasy-reactive-qute] STARTER-CODE SUPPORTED
  Qute integration for RESTEasy Reactive. This extension is not compatible with the quarkus-resteasy extension, or any of the extensions that depend on it.
- RESTEasy Reactive Links [quarkus-resteasy-reactive-links]
  Web Links support for RESTEasy Reactive. Inject web links into response HTTP headers by annotating your endpoint resources.
- REST Client Reactive [quarkus-rest-client-reactive] SUPPORTED
  Call REST services reactively
- REST Client Reactive Jackson [quarkus-rest-client-reactive-jackson] SUPPORTED
  Jackson serialization support for REST Client Reactive
- REST Client Reactive JSON-B [quarkus-rest-client-reactive-jsonb]
  JSON-B serialization support for REST Client Reactive
- REST Client Reactive JAXB [quarkus-rest-client-reactive-jaxb]

What can it offer?

Quarkus' Industry Value

Cost Savings

Low memory footprint, fast startup-time, lesser disk footprint, and serverless deployments

Speed and Agility

Developer productivity, extensions, competitive edge

Standard and Support

Cloud Native Computing Foundation, RedHat, Kubernetes, Eclipse, Jakarta EE

Active community, RedHat build support

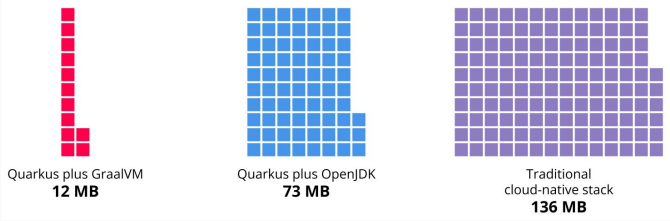
Sustainability

Deploy greener applications, Minimize carbon footprint

Cost Savings

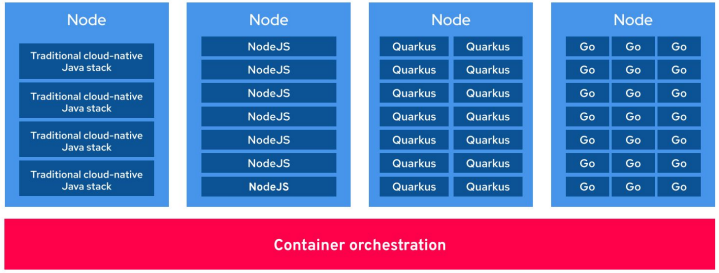
Efficient framework in the cloud

Small Memory Footprint



Fast start-up time

Less Disk Footprint

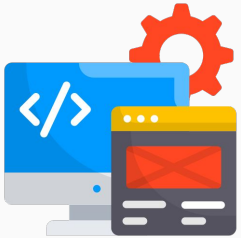


Speed and Agility

Faster time to market

Developers can leverage their existing Java expertise to build modern and cloud-native solutions.

Developer productivity using extensions to create applications quicker



Crafted from well-known libraries such as Spring, Hibernate, Microprofile and Micrometer

Standard and Support

Reliable technology with active community and enterprise support

Quarkus 3.2 is the first LTS release version.

Enterprise build support from RedHat

Active community with regular releases

Kubernetes/Cloud-native technology stack

Performance

Framework Comparison

VS

The Problem

Spring Boot is a well-known and stable framework for creating microservice, event-driven and serverless application.

Unfortunately, it has struggles in memory and startup-times. The root cause of this is due to *Reflection*.

Reflection

Inspect the internal properties of your Java program and modify its behavior upon execution or during runtime.

*java.lang.reflect.**

The Solution

Quarkus uses build-time dependency injection based on CDI 2.0

This is implemented in *Quarkus ArC*.

Metrics

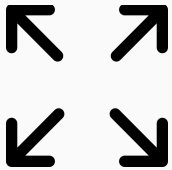
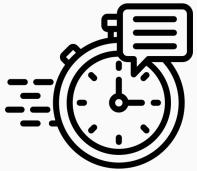


IMAGE SIZE

STARTUP TIME

MEMORY USAGE



RESPONSE TIME

JIT vs AOT

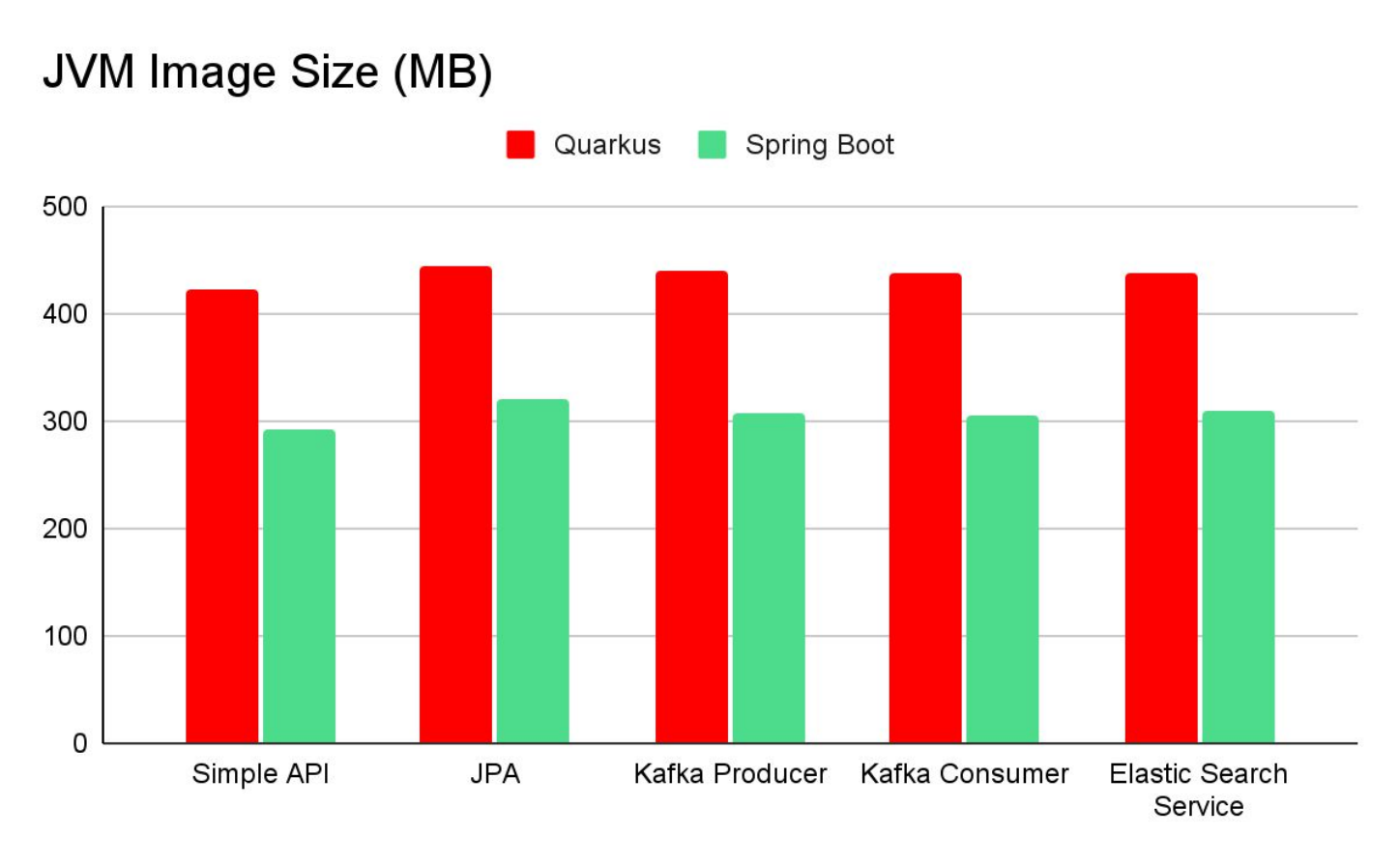
Just in Time compilation

- Default compiler strategy
- Dynamic compilation
- Generate metadata during runtime

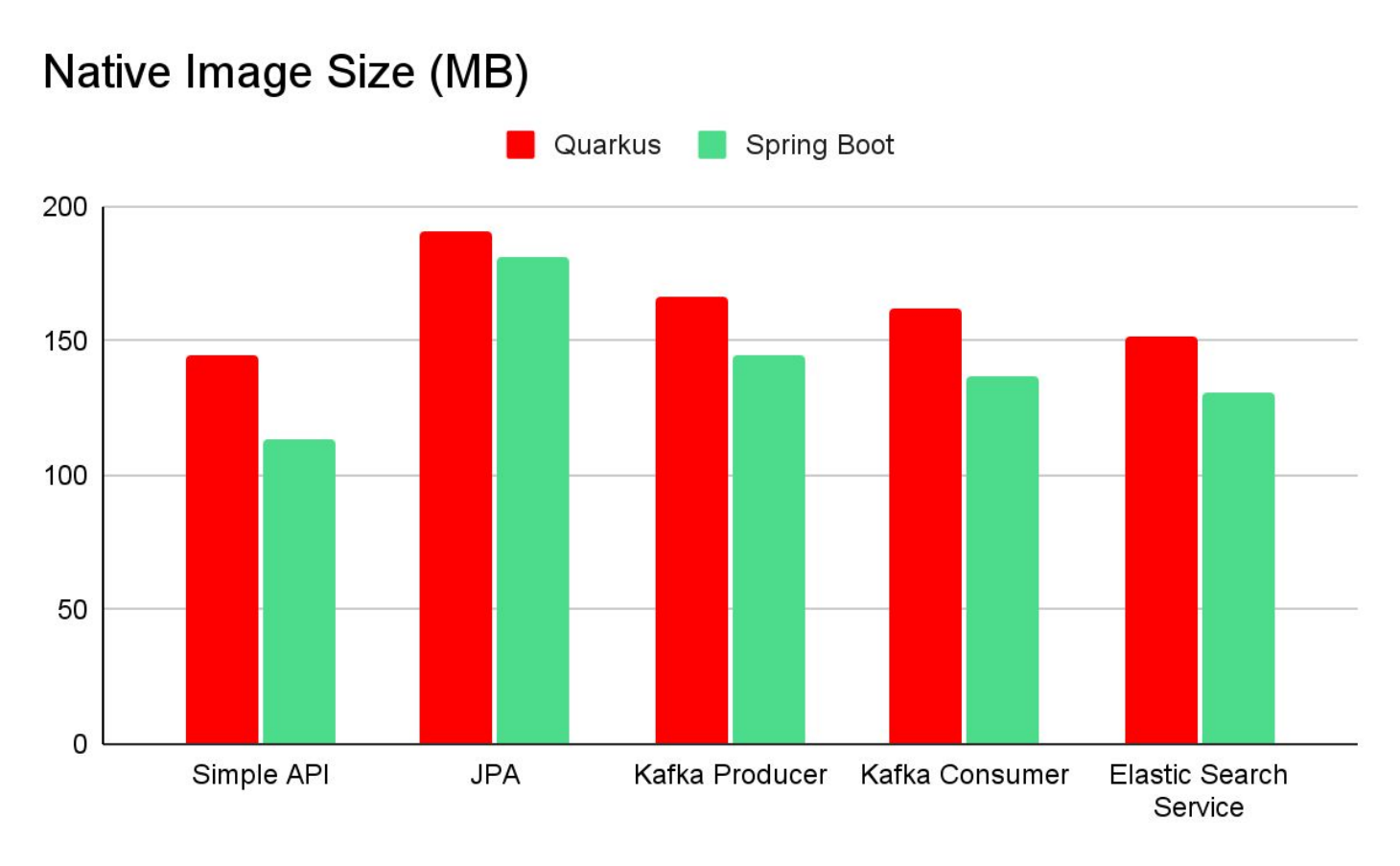
Ahead of Time Compilation

- GraalVM (Native)
- Compile required metadata in build time
- Reduce application size and optimal throughput
- Caveat: GraalVM Supported Libraries

Image Size

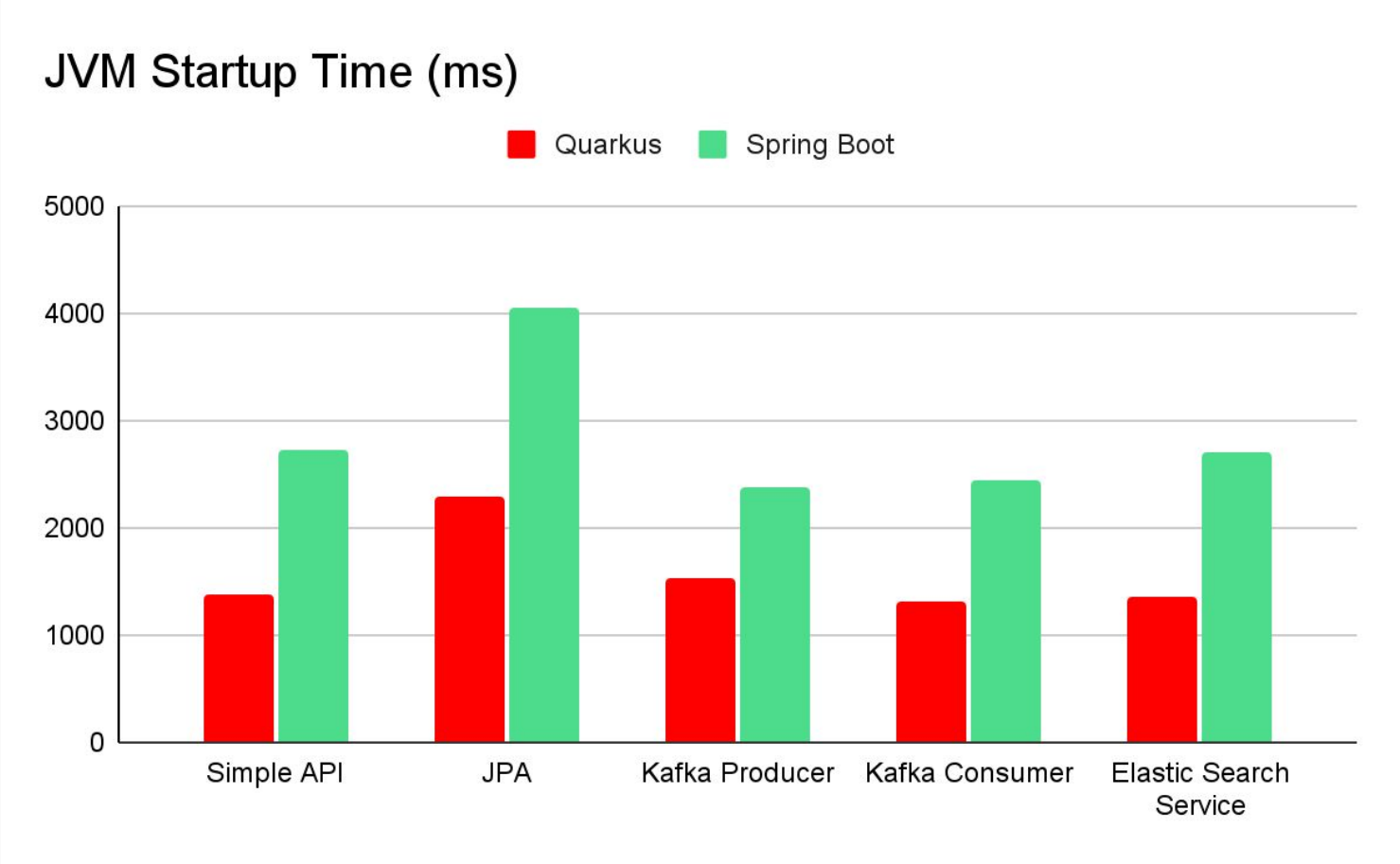


Source: https://github.com/ivangfr/graalvm-quarkus-micronaut-springboot

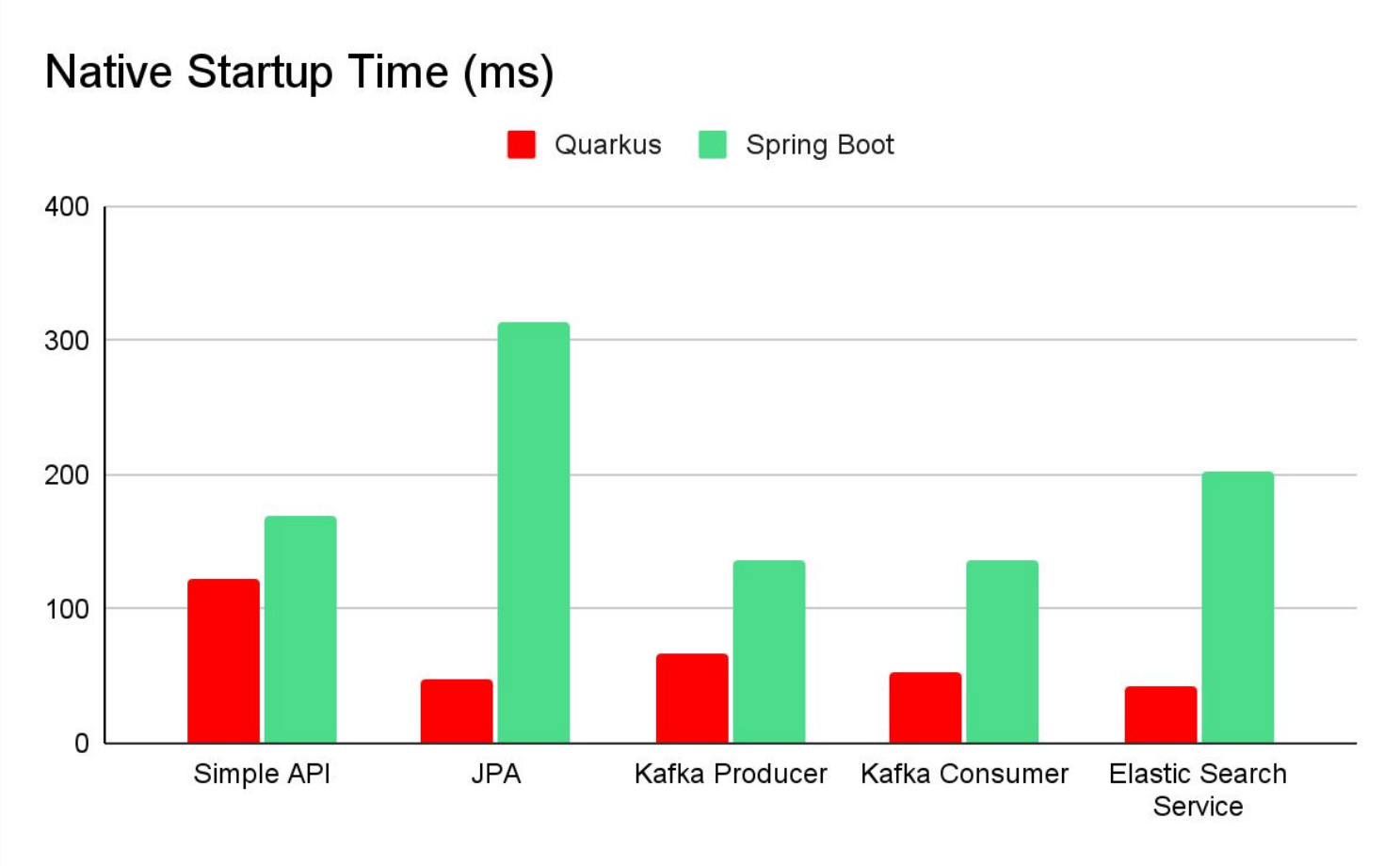


Source:
https://github.com/ivangfr/graalvm-quarkus-micronaut-springboot

Startup Time

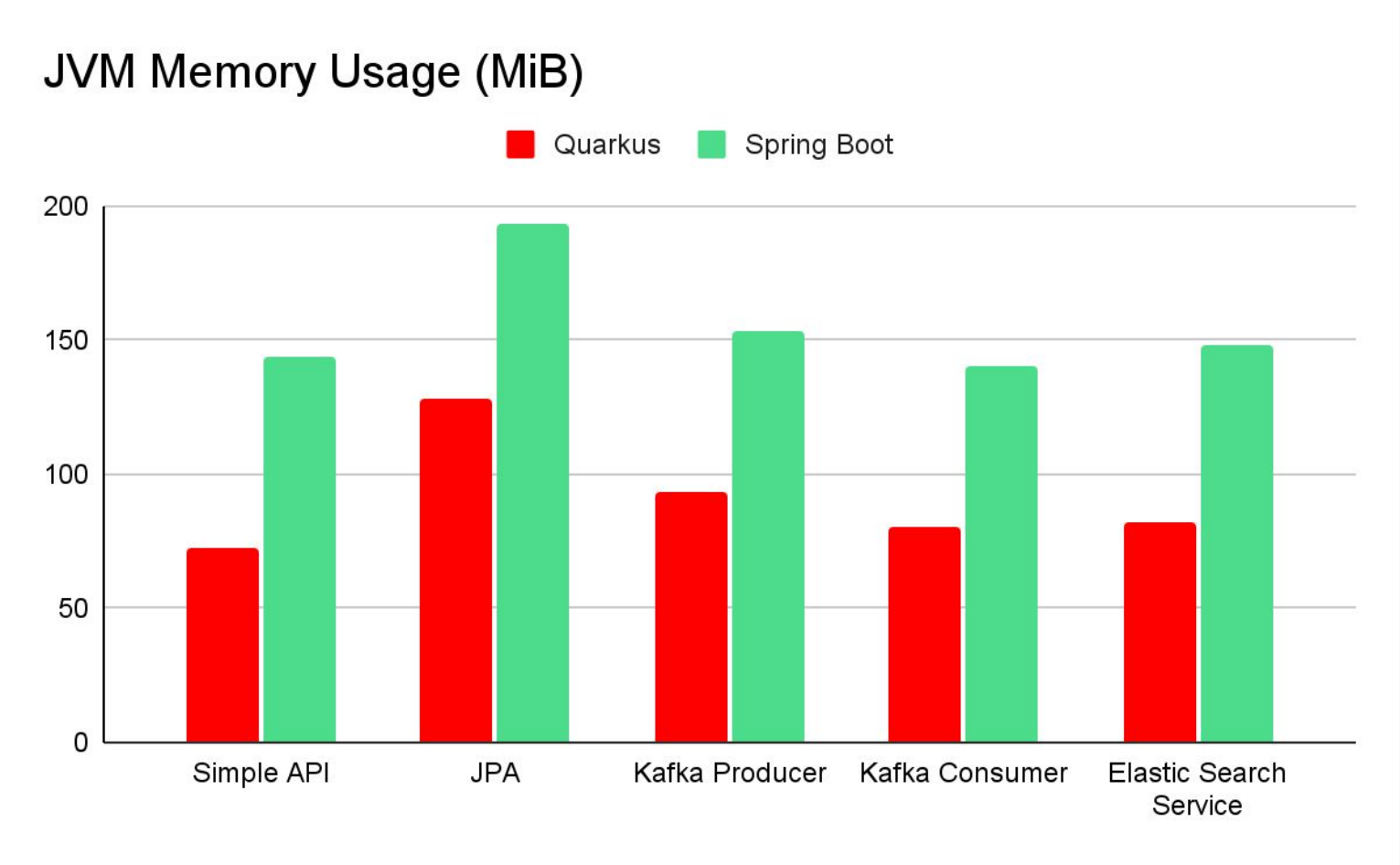


Source:
https://github.com/ivangfr/graalvm-quarkus-micronaut-springboot

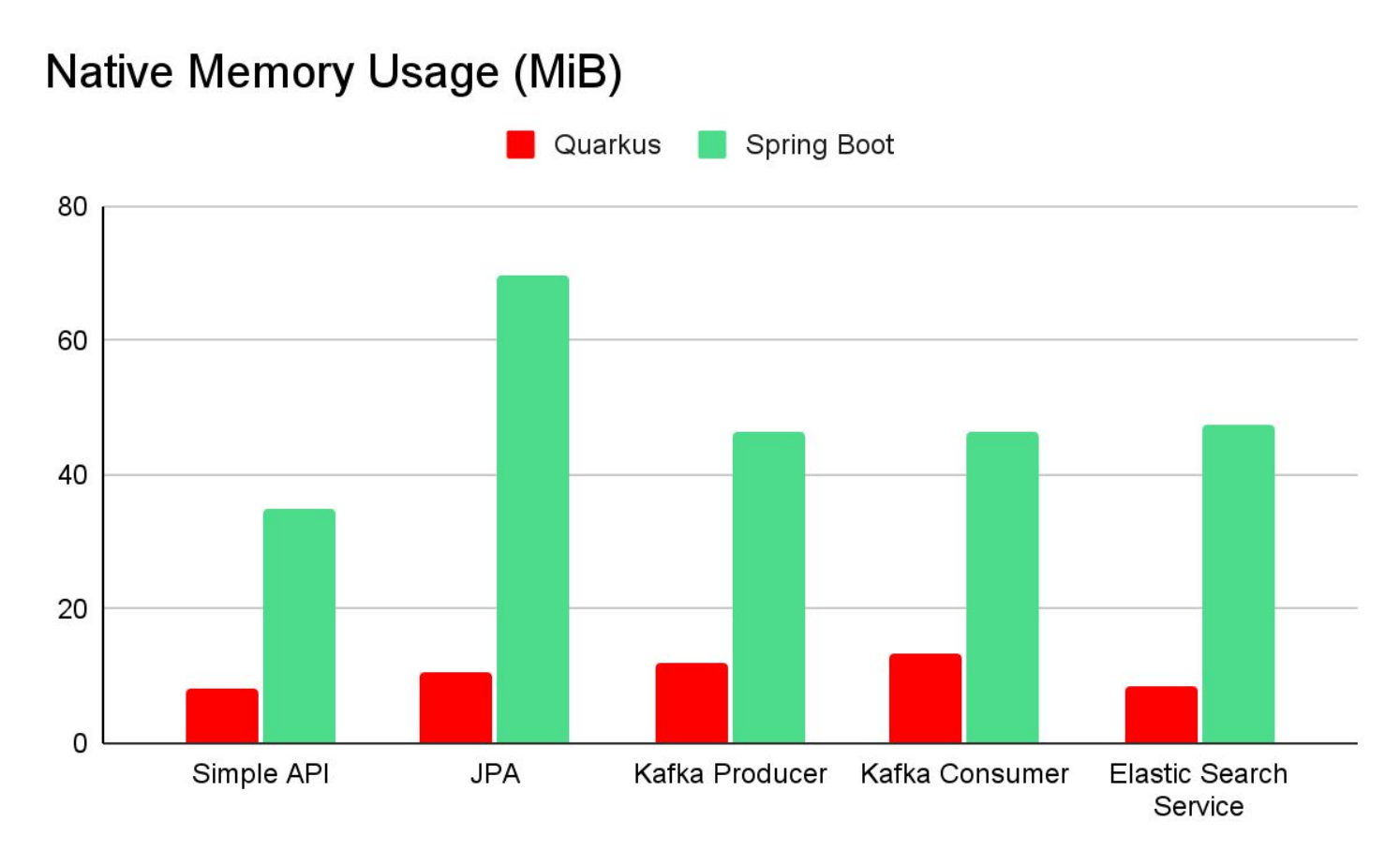


Source:
https://github.com/ivangfr/graalvm-quarkus-micronaut-springboot

Memory Usage



Source:
https://github.com/ivangfr/graalvm-quarkus-micronaut-springboot

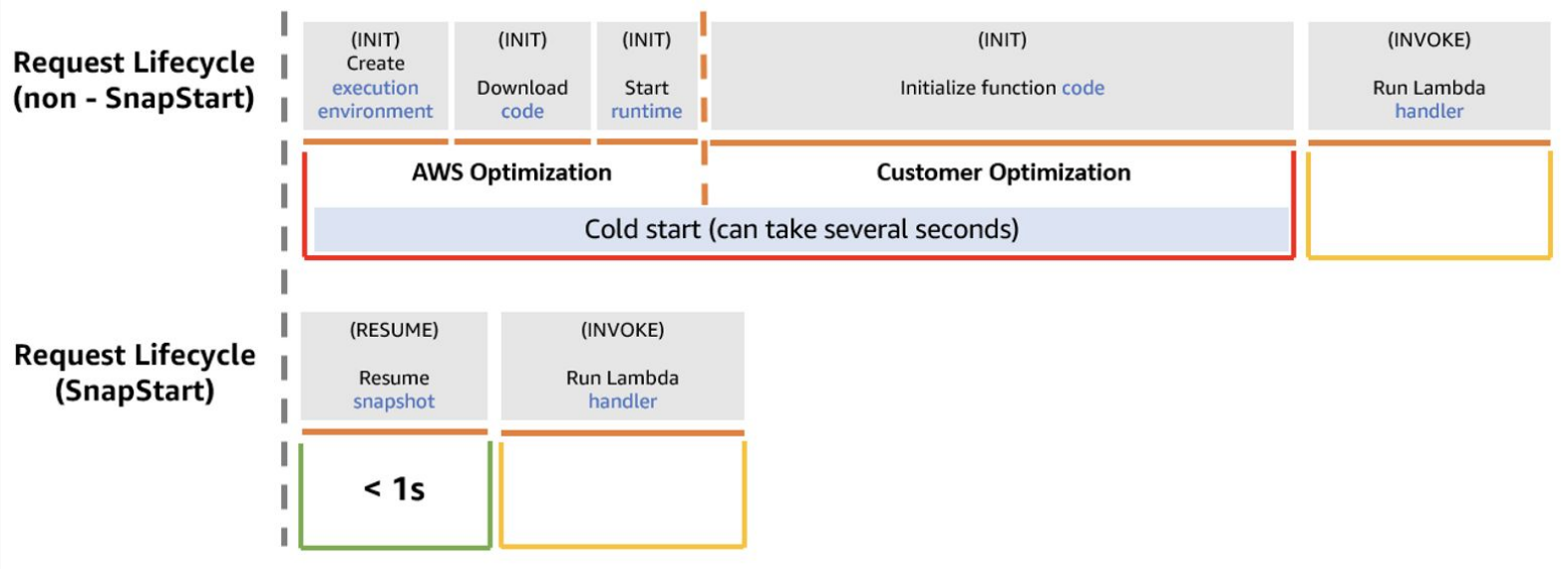


Source: https://github.com/ivangfr/graalvm-quarkus-micronaut-springboot

Experimentation in AWS

Response Time

AWS Lambda SnapStart



Source: [Starting up faster with AWS Lambda Snapstart]

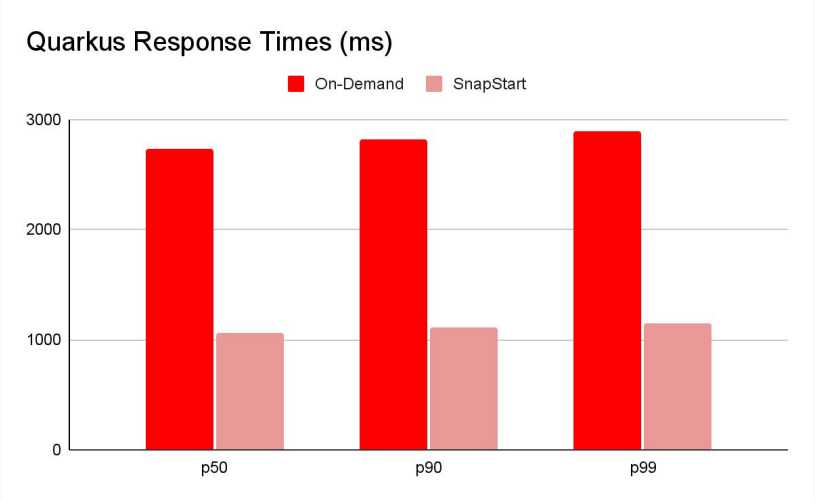
Performance in AWS Lambda

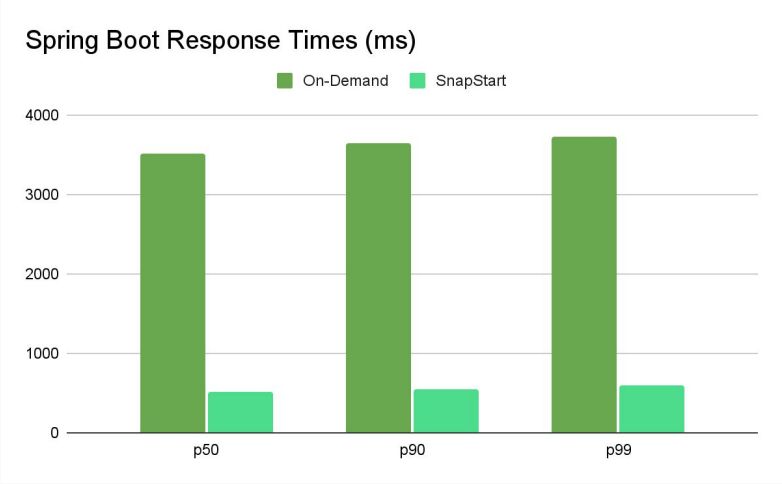
Observe the Cold start performance of Quarkus in On-Demand and SnapStart API Invocation using Quarkus.

The memory used for Lambda performance comparison is 512 MB. It was done with 100 requests with 10 iterations.

Spring Boot had the significant result when using SnapStart with 6x more efficient and 2x for Quarkus.

Source: [Cold Start with SnapStart Comparison]





Spring to Quarkus

Supported Spring features in Quarkus

- Spring Boot Properties
- Spring Cache
- Spring Cloud
- Spring Data & Data Rest
- Spring Dependency Injection
- Spring Scheduled
- Spring Security
- Spring Web

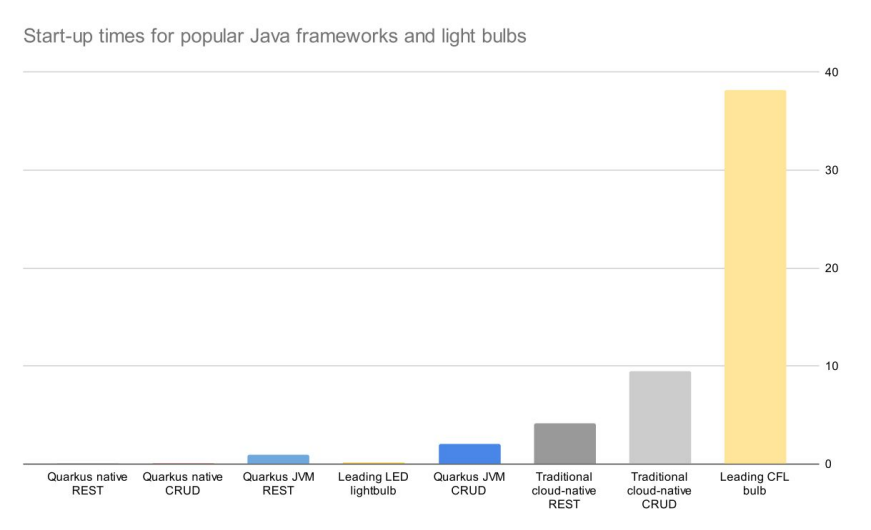
Sustainability

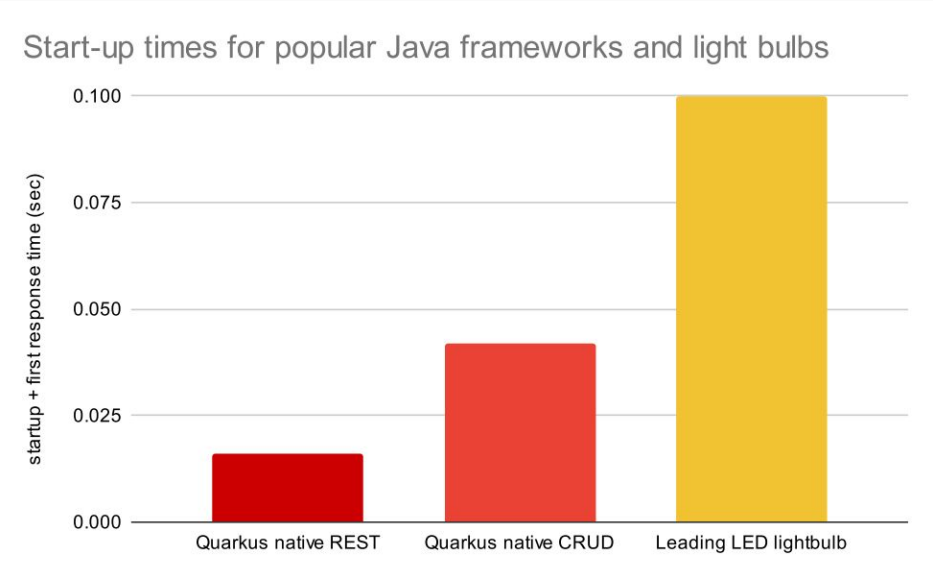
Greener Applications

Quarkus’ impact in the environment

- Efficient handling in energy
- Reduce carbonization
- High Density

Quarkus vs Light Bulb

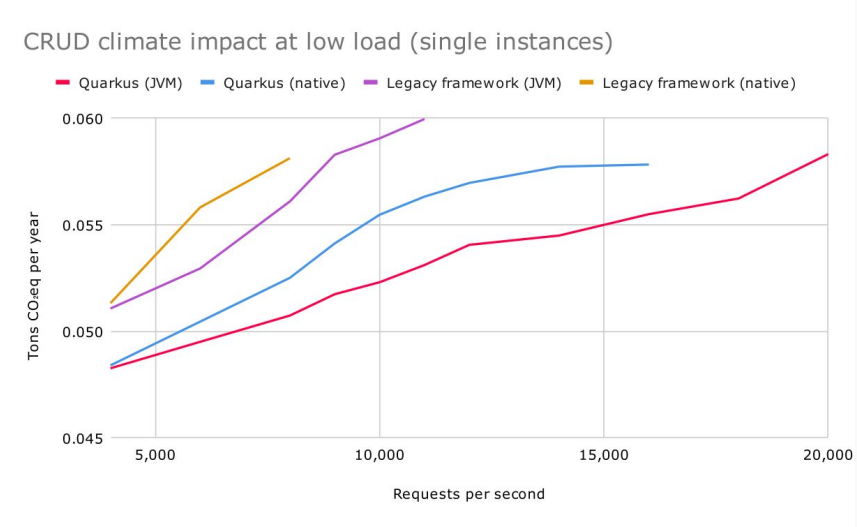


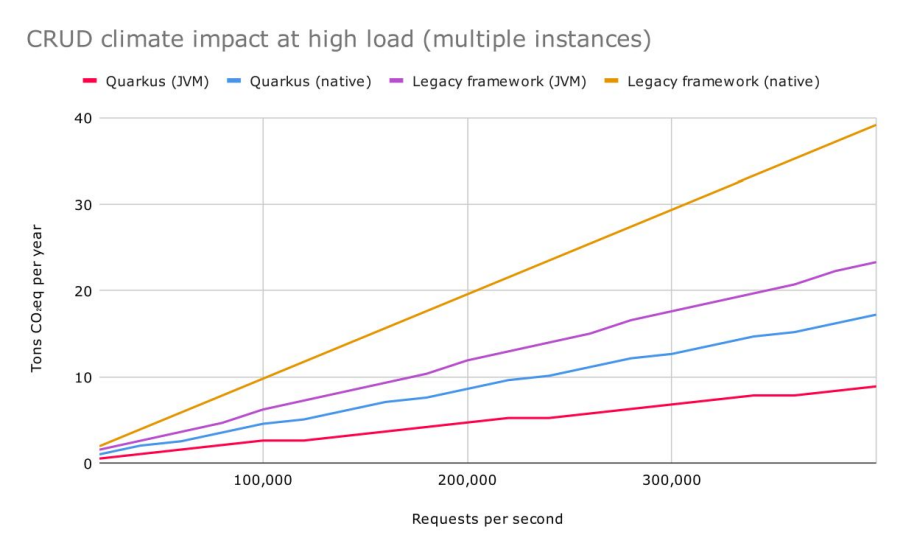


Frameworks: Boot up time plus 1st response time
Light Bulb: up to full brightness

Source: RedHat Greener Application Executive Summary

Power Consumption





The machine was a two CPU machine, with 16 cores per CPU. The application's CPU affinity was set to four specific cores on one CPU, and the application was the only process running on those cores. The heap was not pinned, and could go up to 12G.

The CO2-eq figures are based on the U.S. energy mix.

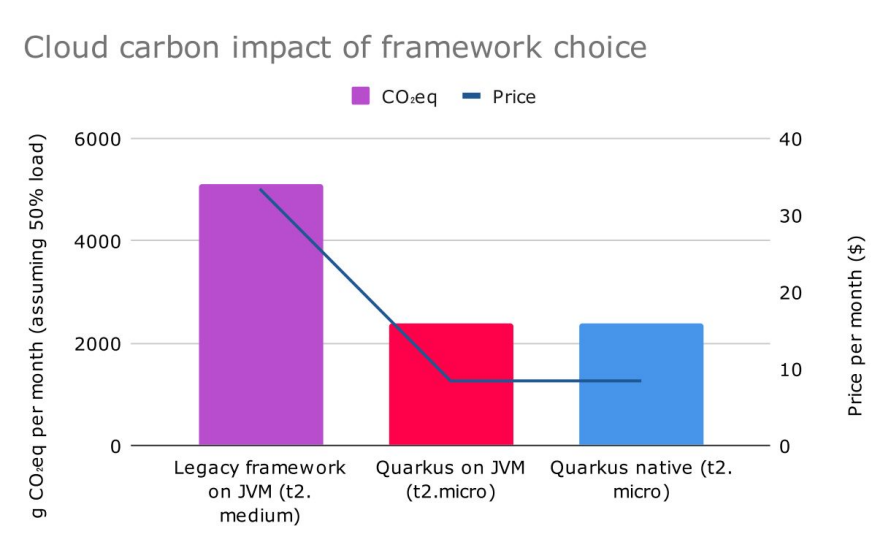
Source: RedHat Greener Application Executive Summary

Density

Quarkus team deployed an application to the cloud, and loaded with 800 requests per minute over 20 days for multiple instances. It assumes using the datacenter, us-east-1 datacenter and a 50% load.

Table 1. Comparing traditional framework on JVM to Quarkus

Framework	Instance Type	Price per hour	CO2-eq per hour
Legacy framework on JVM	t2.medium (2 vCPU, 4GB)	$33.40	5112 g
Quarkus on JVM	t2.micro (1 vCPU, 1GB)	$8.40	2376 g
Quarkus native	t2.micro (1 vCPU, 1GB)	$8.40	2376 g



Source: RedHat Greener Application Executive Summary

Summary

Conclusion

- ➔ Quarkus showed cloud-native characteristics and industry values
- ➔ Quarkus has higher image size but significantly lower in memory usage and startup time compared to Spring Boot
- ➔ Spring Boot was significantly optimized compared to Quarkus in AWS Lambda SnapStart
- ➔ Quarkus has an environmental benefits and significant savings
- ➔ Quarkus 3.2, the first LTS version

It’s up to you to decide if Quarkus is suitable in your use case

Join the Java Community

bit.ly/join-foojay-slack

facebook.com/groups/jugph

meetup.com/java-user-group-ph

twitter.com/jugphilippines

Q & A