#### Agenda

- Discussion
  - Cloud Native
  - GraalVM JIT
  - GraalVM AOT (native image)
- Resources → try it at home / in a sandbox
- Walk through / demo

## "We need to deliver SaaS"

#### **SaaS implementation Plan**

- Virtualise on premises solution
- Host VMs in cloud
- Set up network access
- Sign up customers





6 Copyright © 2022, Oracle and/or its affiliates. All rights reserved.

## "Constructive Laziness"

# "Simplicity – the art of maximizing the amount of work not done – is essential.

Agile Principle #10 http://agilemainifesto.org/principles





# "Use the <u>force</u> platform"

#### **Cloud Economics**



- Minimise infrastructure costs
  - Only run
    - What you need
    - When you need it
- Minimise development costs
  - Only develop what you need to
  - Delegate to the platform when you can



# **CLOUD NATIVE** COMPUTING FOUNDATION

## **CNCF Cloud Native Definition v1.0**

#### Approved by TOC: 2018-06-11

日本語版 (Japanese) | 한국어 (Korean) | Deutsch (German) | Español (Spanish) | (Hebrew) | 中文版本 (Chinese) | (Arabic) | Français (French) | Polski (Polish) | Português Brasileiro (Portuguese-BR) | Português de Portugal (Portuguese-PT) | Русский (Russian) | Bahasa Indonesia (Indonesian) | Тürkçe (Turkish) | Български (Bulgarian) | ไทย (Thai) | Magyar (Hungarian) | Hindi (Indian)

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We democratize state-of-the-art patterns to make these innovations accessible for everyone.

#### **Smaller components**

Serverless, Microservices







# "Runtime Efficiency"

(doing more with less)

# ORACLE Labs

# A decade of GraalVM research @ Oracle

A multilingual VM can be 100% compatible with any language, meeting or beating performance of single language VMs



🛛 🔍 🕒 🔓 GraalVM	× +				
→ C  a graalvm.org/ Waveney-slt.co.uk ★ Bookma	downloads/ arks 📄 OCI-Cert 📄 Unseen University 📄 W	ork 🗎 Chromebook 📄 Code 🗎 Life			Ú 🌣 💽 :
	GraalVM.	Docs Community	Videos Blog	Download	
	Download	GraalVM			
	GraalVM 22.0	GraalVM 21.3	GraalVM 20.3	Developer Builds	
	GraalVM Commun Community supported open source I Release notes	<b>hity 22.0</b>	GraalVM Enter Oracle 24x7 supported comme Release notes	prise 22.0	
	Download	II 🔕 é	Download	<b>=</b> 👌 🗉	
	Docker Images     GraalVM Commun	ity Container Images	→ GraalVM Enter	rprise Container Images	

Ç	Product ~ Solutions ~ Open So	ource Pricing	Search	∑ Sign in S		
G oracle / graal Public				🗘 Notifications		
<> Code   Issues 814	103 Rull requests 103 R Discussions	s 🕞 Actions 🖽 Projects 💿 ए Security 🗠 Insights				
		tags	Go to file Code -	About		
	eregon [GR-26395] Periodic update of the truffleruby import • ad19dd4 2 hours ago 🕚 69,291 commits			GraalVM: Run Programs Faster Anywhere 🖋		
	.devcontainer	Simplify devcontainer config.	4 months ago			
	igithub	moved mx_version from graal_common.json to common.json	3 days ago	javascript ruby python c java		
	includes	Add javadoc under version-specific directory in graalvm-website	7 months ago	vm r polyglot		
	Compiler	[GR-41636] Fix canonicalization of conditional to truncate (GitHub #51	01 5 hours ago	C Readme		
	docs	[GR-41207] Add missing code snippet in Debug Native Executables w	it 18 days ago	্যু View license		
	espresso	Group ci files in each suite	yesterday	் பி.6k stars		
	iava-benchmarks	Quick fix	yesterday			
	negex	Group ci files in each suite	yesterday	<b>೪ 1.4k</b> forks		
	🖿 sdk	[GR-41577] Remove GuardedAnnotationAccess and DirectAnnotation	15 hours ago			
	substratevm	[GR-41019] Always run phaseplan verification.	9 hours ago	Contributors 250		
	sulong	Quick fix	yesterday	📵 🛞 🚱 🏟 🊱		
	tools	Group ci files in each suite	yesterday	🕞 🤤 🎲 🥵		
	🖿 truffle	Group ci files in each suite	yesterday	+ 239 contributors		

#### **GraalVM Enterprise Edition**



- Included with OCI
- Production Support
- Higher performance, more advanced than CE

#### **Renaissance Suite** (higher is better)



JVM implementation OpenJDK 17 GraalVM CE 22.2 JDK 17 GraalVM EE 22.2 JDK 17

#### Most of Graal Is Developed in Java



package org.graalvm.compiler.phases.common;

public class ConditionalEliminationPhase extends BasePhase<CoreProviders> {

protected void run(StructuredGraph graph, CoreProviders context) {
 try (DebugContext.Scope s = graph.getDebug().scope("DominatorConditionalElimination")) {
 BlockMap<List<Node>> blockToNodes = null;
 NodeMap<Block> nodeToBlock = null;
 ControlFlowGraph cfg = ControlFlowGraph.compute(graph, true, true, true, true);
 if (fullSchedule) {
 if (moveGuards && Options.MoveGuardsUpwards.getValue(graph.getOptions())) {
 cfg.visitDominatorTree(new MoveGuardsUpwards());
 }
 }
}

••••

Simpler programming model, Java ecosystem, debugging tools

## Faster progress!

#### Most of Graal's Tooling Is Developed in Java



#### **Oracle GraalVM Enterprise Edition**





### What is GraalVM Enterprise?

A distribution of Oracle JDK with the high performance GraalVM optimizing compiler that provides significant improvements in application speed and efficiency



High-performance optimizing Justin-Time (JIT) compiler



Ahead-of-Time (AOT) "Native Image" generator



Multi-language support for the JVM



## **GraalVM Enterprise**

Just-in-time compilation

#### GraalVM Enterprise JIT Compiler—Ideal for traditional Java workloads



#### GraalVM Enterprise: Oracle JDK + GraalVM JIT Compiler



#### **GraalVM Enterprise**—**Faster**

Increased performance in real-world application benchmarks



GraalVM 22.0 JDK 11 JIT vs. C2 JIT

GraalVM JIT Speedup

#### **GraalVM Enterprise**—**Faster**

Increased performance in real-world application benchmarks



#### GraalVM 22.0 JDK 17 JIT vs. C2 JIT

GraalVM JIT Speedup

## GraalVM Enterprise—Faster Scala

Scalabench Up to 2.75x faster on GraalVM Enterprise





## **GraalVM Enterprise**—**Leaner**

Higher performance with less memory



RUNNING TIME VS MEMORY, NAIVE-BAYES, JDK 11 (LOWER IS BETTER)

> On the Renaissance "naïve-bayes" benchmark, GraalVM Enterprise outperforms OpenJDK 11—regardless of the amount of available RAM.

#### GraalVM Enterprise with 6GB outperforms OpenJDK with 10GB

Source: https://blogs.oracle.com/graalvm/apache-spark%e2%80%94lightning-fast-on-graalvm-enterprise

## Seeing Escape Analysis Working

Chris Seaton, 16 December 2020



This article originally appeared in the Java Advent 2020.

You may have heard of a compiler analysis phase called *escape analysis*. It informs an optimisation called *scalar replacement of aggregates* that removes unnecessary allocation of Java objects. I find that people often have a misunderstanding of what this optimisation really does and an under-appreciation of what it's capable of. We can know it better by seeing it working in practice.



**Execution Time** 

Why is GraalVM Enterprise Faster?



**Execution Time** 



# DØLLEMC

"Oracle GraalVM Enterprise Edition was the performance choice for our Dell EMC Servers. Java workload analysis and SPECjbb®2015 benchmark improving max-jOPS results by almost 8%."

#### **Kurtis Bowman**

Director of Architecture, Server Office of the CTO





"We save a lot of money and CPU cycles"

# 10% performance increase20% reduction in latency

**Chris Thalinger** Staff System Engineer, Twitter

### **GraalVM Enterprise in Oracle Cloud Infrastructure**

Real-world example



## 10% improvement in transactions/sec

## **0 ISSUES** 30+ Million core hours

GraalVM Powers Oracle Cloud Infrastructure



## No code changes.

## No code changes.

\$ java -cp app.jar my.package.Main



## **GraalVM Enterprise**

Ahead-of-time compilation



凸 LIKE

by

13

reviewed by



G 4  $\square$ Ō. Key Takeaways MAY 19, 2022 • 19 MIN READ := Spring Boot 3 and Spring Framework 6, due in late 2022, will have built-in support for native Java. Josh Long FOLLOW • For Spring Framework 5.x and Spring Boot 2.x users, Spring Native is the way to Spring Developer Advocate at Tanzu, a division of VMware go. Spring Native provides integrations for a vast ecosystem of libraries. Karsten Silz FOLLOW But Spring Native also ships a component model that allows you to extend native Full-Stack Java Developer & Contractor compilation support for other libraries. GraalVM AOT compilation offers a lot of possibilities with some (negotiable) Write for InfoQ costs.



WA

WASM

Watch

Watch

MAY 16, 2022 youtube.com

MAR 22, 2022 youtube.com

## GraalVM Enterprise Native Image—Ahead-of-time compiler & runtime

Microservices and Containers



## Up to 5x less memory 100x faster startup

## No code changes.

\$ native-image -cp app.jar my.package.Main

\$./main

46 Copyright © 2022, Oracle and/or its affiliates. All rights reserved



#### **Closed World Assumption**

- native-image assumes it knows everything
- Aggressively eliminates dead code
- Points to (static) analysis needs to see all bytecode
  - Eagerly loads referenced classes
- No loading of new classes at runtime
- Dynamic parts of Java require build time "reachability metadata" (configuration)
  - reflection, proxies, resources, JNI
  - Tooling for this (finding and telling)
  - <u>Github repo</u> for libraries



#### **Reachability metadata**



```
"condition": {
   "typeReachable": "com.zaxxer.hikari.util.ConcurrentBag"
  },
  "name": "[Lcom.zaxxer.hikari.util.ConcurrentBag$IConcurrentBagEntry;"
 },
  "condition": {
   "typeReachable": "com.zaxxer.hikari.pool.PoolEntry"
  },
  "name": "[Ljava.sql.Statement;"
...
```

**GraalVM Enterprise Native Image** 

# Supported by microservice frameworks and platforms







**CLOUD NATIVE** COMPUTING FOUNDATION

## Go Native with Spring Boot and GraalVM



MAY 19, 2022 • 19 MIN READ

#### **Key Takeaways**

support for native Java.

by



FOLLOW

Spring Developer Advocate at Tanzu, a division of VMware

reviewed by



Full-Stack Java Developer & Contractor

Write for InfoQ

For Spring Framework 5.x and Spring Boot 2.x users, Spring Native is the way to

Spring Boot 3 and Spring Framework 6, due in late 2022, will have built-in

- Spring Native provides integrations for a vast ecosystem of libraries.
- But Spring Native also ships a component model that allows you to extend native compilation support for other libraries.
- GraalVM AOT compilation offers a lot of possibilities with some (negotiable) costs.



#### Not a silver bullet!



- Closed world assumption
  - Tracing / manual configuration may be required
  - Reflection, dynamic class loading
- Retrofitting to existing code bases can require some work
- Compilation is not a security / obfuscation tool!
  - But there are security benefits (think log4j)
- Better for new development
  - Serverless
  - Microservices

#### **Native Image and Cloud Native**

- Smaller runtime
  - Big jar → small binary
- Smaller containers
  - "distroless" → smaller attack surface
- Faster startup time
  - Initialisation during AOT compilation
  - Lower latency
- Lower memory usage



## GraalVM Enterprise Native Image—Java productivity with C-like performance

Microservices and Containers



Lower compute requirements and faster execution reduces infrastructure/cloud costs

## **GraalVM Enterprise Native Image—drastic resource reductions**

#### **Microservices and Containers**



Microservice on GraalVM Enterprise (JIT mode)

GraalVM Enterprise Native Image generated native executable

## **GraalVM Native Image—Best Solution for Containerized Java**

#### Microservices and Containers



Lower compute requirements and faster execution reduces infrastructure/cloud costs

## How fast is your PetClinic?

Sample	On the JDK		native-executable	native-executable			
petclinic-jdbc	Build:	9s	Build:	194s	+2050%		
	Memory(RSS):	417M	Memory(RSS):	101M	-75%		
	Startup time:	2.6s	Startup time:	0⊾158s	-94%		







Example—Spring Boot with GraalVM Native Image

On the JDK		native-executable		
Build:	9s	Build:	194s	+2050%
Memory(RSS):	417M	Memory(RSS):	101M	-75%
Startup time:	2.6s	Startup time:	0⊵158s	-94%

#### **GraalVM Enterprise throughput**



Popular Framework Benchmark



# Scaffolding

**Graal Cloud Native** 

# **GraalVM**...

## **Graal Cloud Native**

Making it easy to build cloud-portable applications that leverage powerful platform provided managed services using *Spring, Micronaut, Helidon,* and more!



#### **Graal Cloud Native**



Graal Cloud Native lets you take full advantage of powerful cloud services without coupling to proprietary platform APIs.

Graal Cloud Native solves the problem of the lack of standard cross-cloud APIs and makes application portability possible.

Use Graal Cloud Native to leverage services like objectstorage, monitoring, authentication, secret management, and deploy to popular cloud platforms.



#### **Graal Cloud Native** VS Code Tools for OCI

 Powerful VS Code extensions that make it easy to onboard Java developers onto OCI

ථ

- Full support for DevOps projects including Git source control, build pipelines, container repositories, etc. from within VS Code
- OCI Application Dependency Manager integration to detect critical vulnerabilities during application development
- Compatible with OCI Cloud Editor and Cloud IDE

LORER ····	! build_spec.yaml ∪ ●	ជា
-MICRONAUT 다 단 한 의 github i eploy nages re m mshop micronaut.mushop:mushop	<pre>! build_spec.yaml 1 version: 0.1 2 component: build 3 timeoutInSeconds: 20000 4 runAs: root 5 shell: bash 6 env: 7 variables:</pre>	P*- Territoria T
for the second sec	8       "JAVA_HOME" : "/usr/lib64/graalvm/graalvm22-ee-java17"         9       steps:         10       - type: Command         11       name: "Pre-yum repo checks"         12       command:           13       printf "\n\n Contents of graal_spec.yaml file:\n \$(cat graal_spec.yaml) \n\n"         14       printf "Contents of /etc/yum.repos.d/:\n \$(ls -alh /etc/yum.repos.d/) \n"         15       - type: Command         16       - type: Command	
AL CLOUD ☐ mushop (MyTenancy-PHX/HelloComp)	17       name: "Temporary workaround: Set up the yum repo"         18       command:           19       (echo -e '[ol7_oci_included]' && \         20       echo 'name=Oracle Software for OCI users on Oracle Linux \$releasever (\$basearch)' && \         21       echo 'baseurl= <u>https://yum-phx.oracle.com/repo/OracleLinux/OL7/oci/included/\\$basearch/' &amp;&amp; \         22       echo 'gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle' &amp; \         23       echo 'gpgcheck=1' &amp;&amp; \         24       echo 'enalted=1') &gt; /etc/yum.repos.d/oci-included-ol7.repo         25       yum-config-managerenable ol7_optional_latest   </u>	
⊕ container images	26       - type: Command         27       name: "Post-yum repo checks"         28       command:           29         printf "Contents of /etc/yum.repos.d/:\n \$(ls -alh /etc/yum.repos.d/) \n"         30       - type: Command         31       name: "Install GraalVM Enterprise 22.1 Java 17 - JDK and Native Image"         32       command:           33         yum - y install graalvm22-ee-17-native-image         34       - type: Command         35         yum - y install graalvm22-ee-17-native-image	
ERNETES	35 name: "Set the PATH here. JAVA_HOME already set in env > variables above."	
⊖ ⊗0 <u>∧</u> 0	Ln 8, Col 59 Spaces: 2 UTF-8 LF Y	AML 🖉 🕻



#### **Graal Cloud Native** Benefits



Simplified service usage eliminates the need to learn proprietary platform APIs.

Advanced compile-time validation to reduce dev/test/debug cycle. Easily Leverage Cloud Platform Services

Inject cloud services right into an application exactly where they are needed. All major services supported out of the box.

#### Apps start fast and use fewer resources

Out-of-the-box compatibility with GraalVM Native Image let's applications start up to 100x faster than when running on the JVM, deliver immediate peak performance, and require less memory and CPU.



Deploy to different cloud providers with no code changes.

Support for all the major cloud platforms.

← → C developer.oracle.com/java/graalvm/cloud-native/

🗎 Waveney-slt.co.uk 🛧 Bookmarks 🗎 OCI-Cert 🗎 Unseen University 🗎 Work 🗎 Chromebook 🗎 Code 🗎 Life



#### Benefits of Graal Cloud Native



#### Developer Productivity

Save time with out-of-the-box integrations and compile-time validation with common developer technologies.

Building Apps with Micronaut® (20:37) Code on GitHub



Easily Leverage Oracle Cloud Services

Micronaut<sup>®</sup> can inject Oracle Cloud services right into an application exactly where they are needed.

Leverage Oracle Cloud Services (3:53) Code on GitHub

#### æ

Start Fast with Fewer Required Resources

Graal Cloud Native applications compiled by GraalVM Native Image start up to 100x faster than when running on the JVM, operate at peak performance immediately, and require less memory and CPU.

Build Performant Apps (3:03) Code on GitHub

### $\bigotimes$

Deploy to different cloud providers with no code changes.

Multi-Cloud by Design (13:13)

Multicloud by Design

Code on GitHub

#### Get Started with Graal Cloud Native

合☆ 🖙 🗯 🔲 🙆

#### Resources



- Home page <u>https://www.graalvm.org/</u>
- Twitter <u>https://twitter.com/GraalVM</u>
- Medium <u>https://medium.com/graalvm</u>
- Luna Labs
  - <u>https://luna.oracle.com/</u>
  - Use Oracle login (not OCI), search for GraalVM
- Live Labs
  - <u>http://bit.ly/golivelabs</u>
- "GraalVM for Dummies" ebook
  - <u>https://go.oracle.com/LP=105746</u>

#### **Summary**



- With cloud, you pay for what you use
- You need to be as efficient as possible
- Run your Java applications more efficiently with GraalVM EE
- GraalVM EE is included with OCI
- The OCI price performance advantage is even bigger for Java apps
- Try GraalVM EE & OCI today

# Let's head to the <u>lab</u>...

## Thank you

Any Questions?



