



【ハンズオン】 次世代Java高速実行基盤GraalVM

Jun Suzuki

Java Value Solutions Consultant

Java Global Business Unit

日本オラクル株式会社

May 27, 2022

Agenda

- 
1. GraalVM概要とハンズオン内容紹介 (20分)
 2. ハンズオン
 1. 事前準備 (10分)
 2. 演習1. GraalVMのインストール (10分)
 3. 演習2. 高性能JITコンパイラ (10分)
 4. 演習3. Native Image (10分)
 5. 演習4. GraalVMとJavaフレームワーク (15分)
 6. 演習5. GraalVMとDockerコンテナ (15分)

GraalVM Enterpriseとは

100% Oracle JDK互換の次世代多言語プログラミング実行環境



Oracle JDKをベースに開発された次世代Java & 多言語実行基盤

- 高機能なJust-In-Timeコンパイラ
- マイクロサービスに最適なNative Image
- 多言語対応

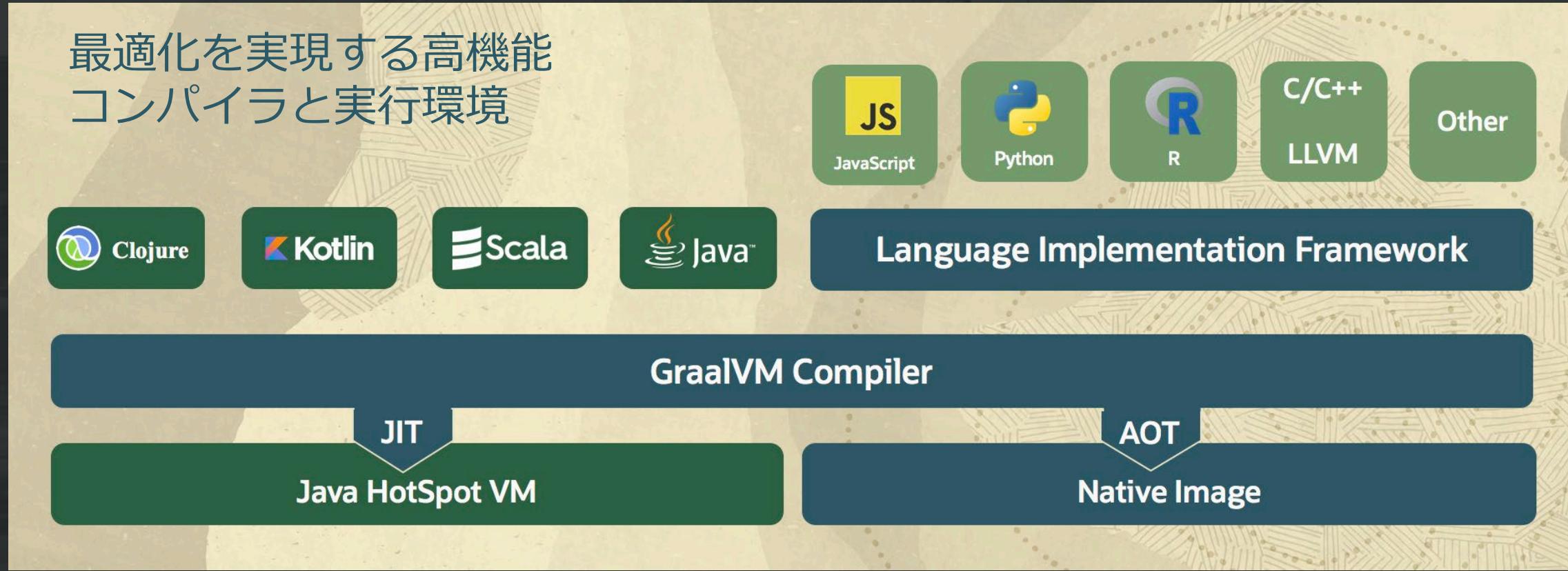
Oracle JDKと互換性を持ち、LTS版Java8/11/17に対応

- 既存JavaアプリケーションがそのままGraalVM上で稼働
- Oracle JDKが4半期ごとにリリースするセキュリティパッチ（CPU）と完全な同期を取る

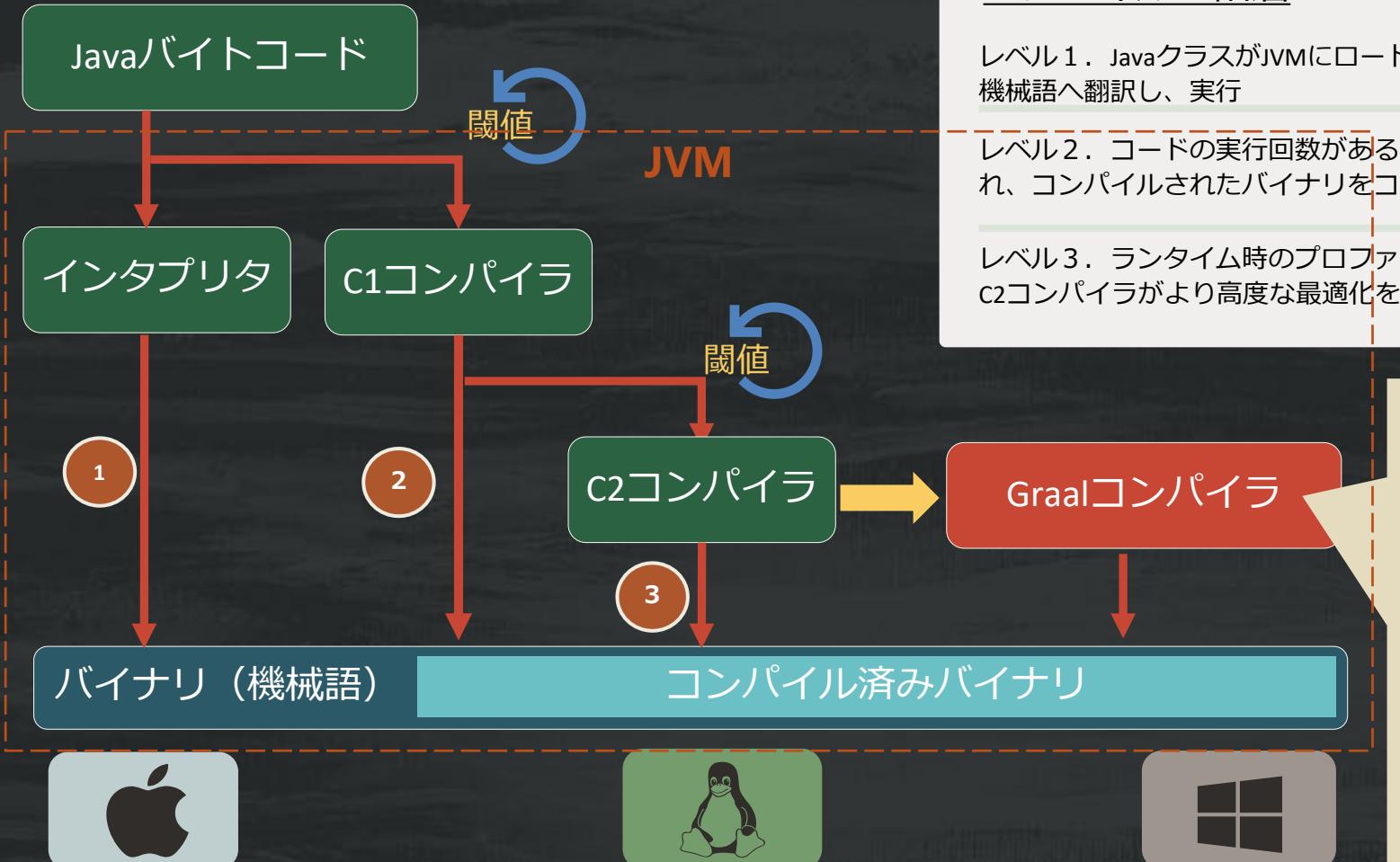
GraalVM EnterpriseはJava SE Subscriptionのライセンスに含まれる

- GraalVM Enterpriseの別料金が発生しない
- NFTCライセンスには含まれない

GraalVM Enterprise アーキテクチャ



高性能JITコンパイラ



コンパイルの階層

レベル1. JavaクラスがJVMにロードされた後、インタプリタがソース・コードを一行づつ機械語へ翻訳し、実行

レベル2. コードの実行回数がある閾値を超えた場合、C1コンパイラによる最適化が行われ、コンパイルされたバイナリをコードキャッシュに保存

レベル3. ランタイム時のプロファイリングにより呼び出し経路が解析、特定され、C2コンパイラがより高度な最適化を実施



◆ インライン化

- getter/setterメソッドの代わりに、実際の変数を特定し代入することにより、メソッドコールの回数を減らし、オーバヘッドを軽減

◆ Escape解析

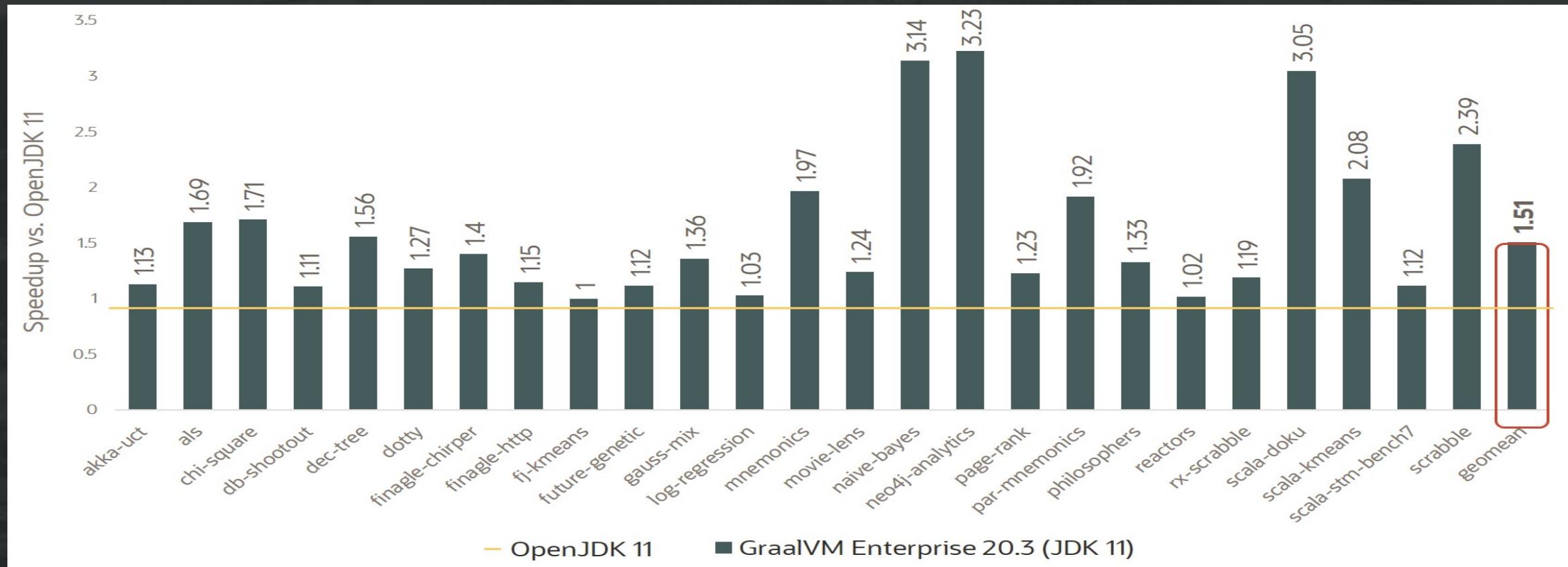
- 変数の適用範囲を特定し、Synchronization作業を最低限に抑え、またローカル変数オブジェクトをJVMヒープからスタックに移動することで、JVMの使用メモリーを減らす

◆ DeOptimization

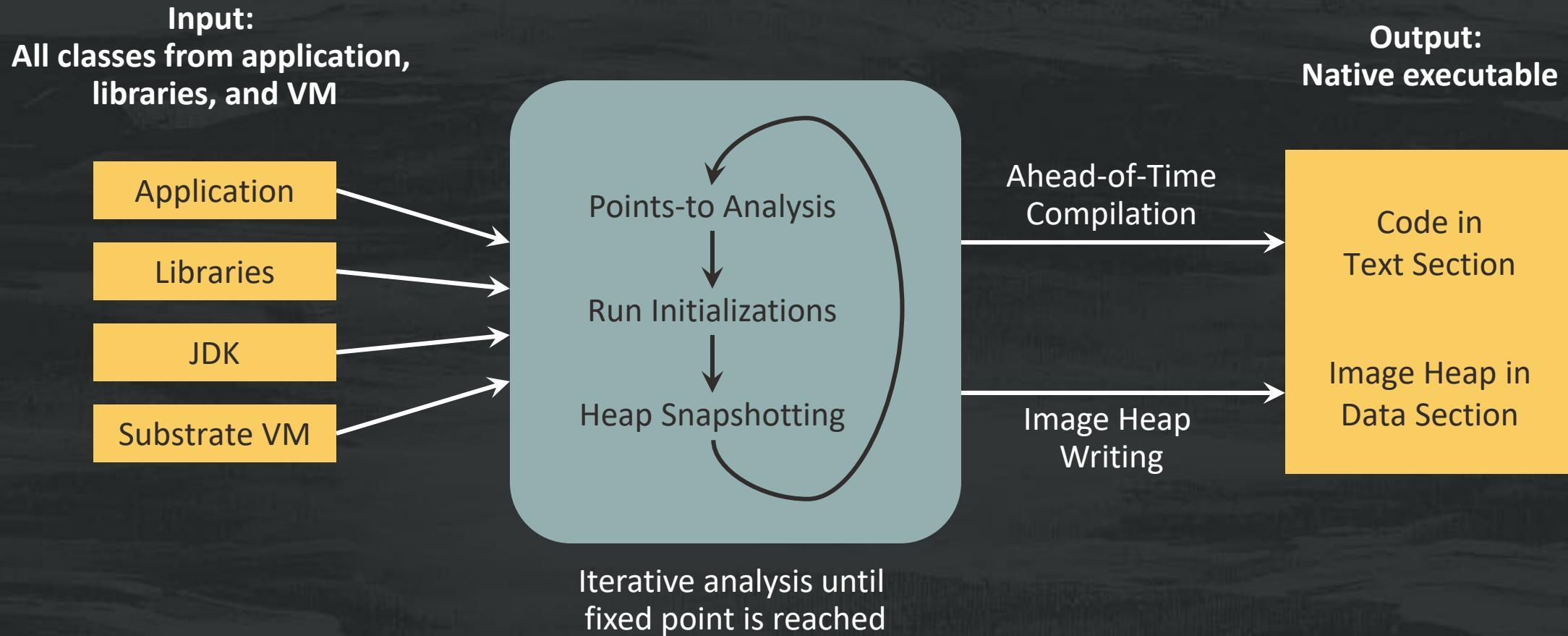
- 最適化対象クラスを自動的に見直すことによりパフォーマンスを高め、また不要となるオブジェクトを特定してキャッシングから除去することによりメモリー負担を軽減

GraalVM EE 20.3のベンチマーク結果

- 最適化されたJITコンパイラ
- 既存インフラからより高いパフォーマンスを引き出す
- OpenJDKと比較して、平均して1.5倍のパフォーマンス向上を実現

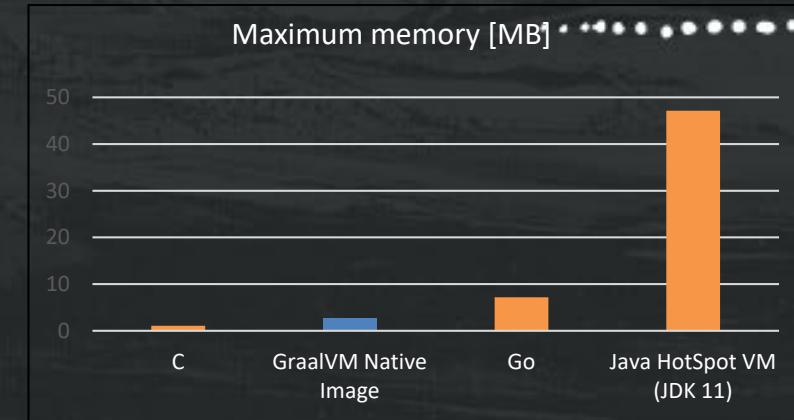


Native Image

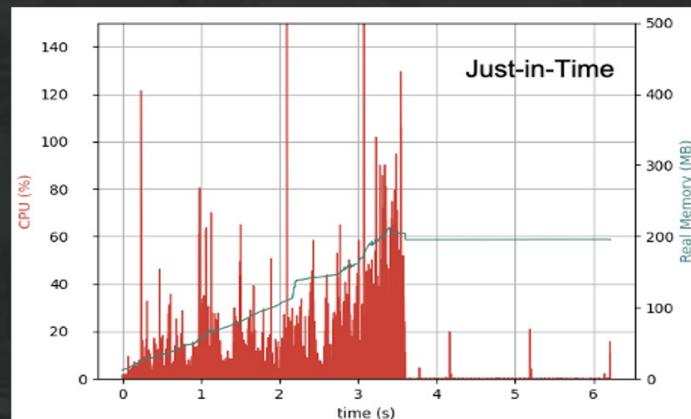


Native Imageのパフォーマンス

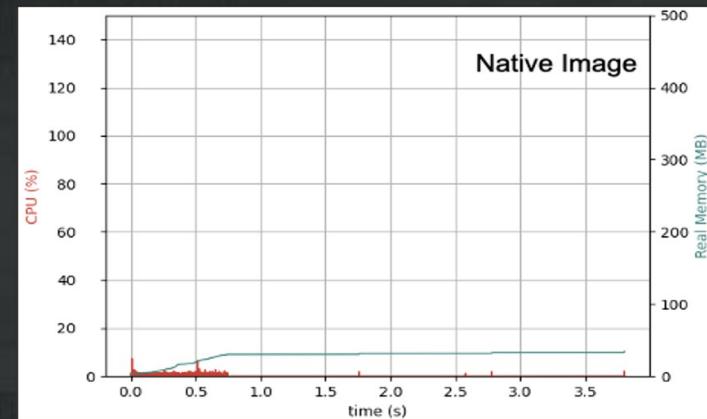
- Native Imageの実行時間とメモリ消費



- コンテナー稼働時のリソース比較：JIT vs AOT



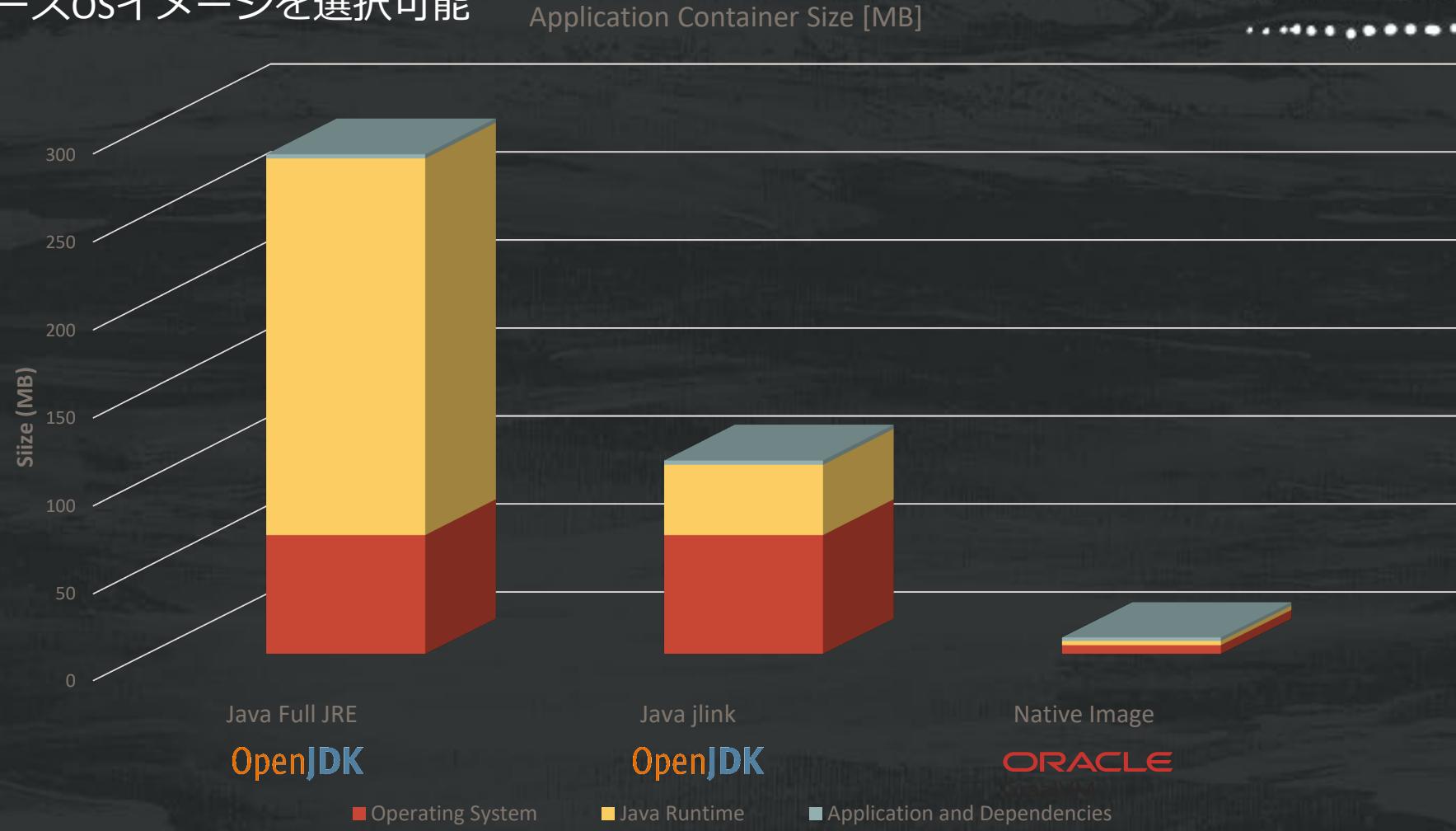
Microservice on GraalVM Enterprise JIT mode



Microservice on GraalVM Enterprise Native Image

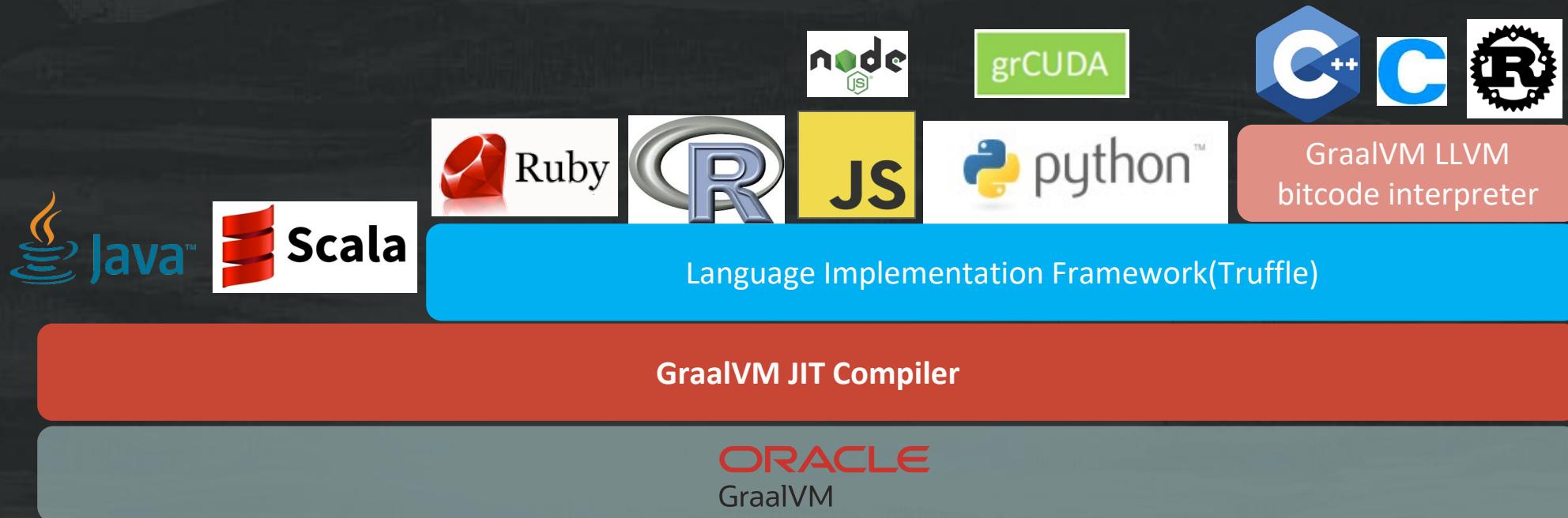
Native Imageとコンテナ

- 実行時Javaランタイムに依存しない
- 軽量なベースOSイメージを選択可能



多言語プログラミング対応

- Java以外の言語のインタープリターを実装するためのフレームワークを提供
 - フレームワーク経由でGraalVMのJITやGCを利用可能
- JavaScript, R, Ruby, Python, LLVM, Web Assemblyなど各言語の実行環境を提供
 - 1つのアプリケーションの中、複数言語の相互運用性を提供
 - 例：JavaScript のコードからRubyのメソッドを呼び出せる
 - Rubyのランタイム起動は不要



多言語プログラミング対応

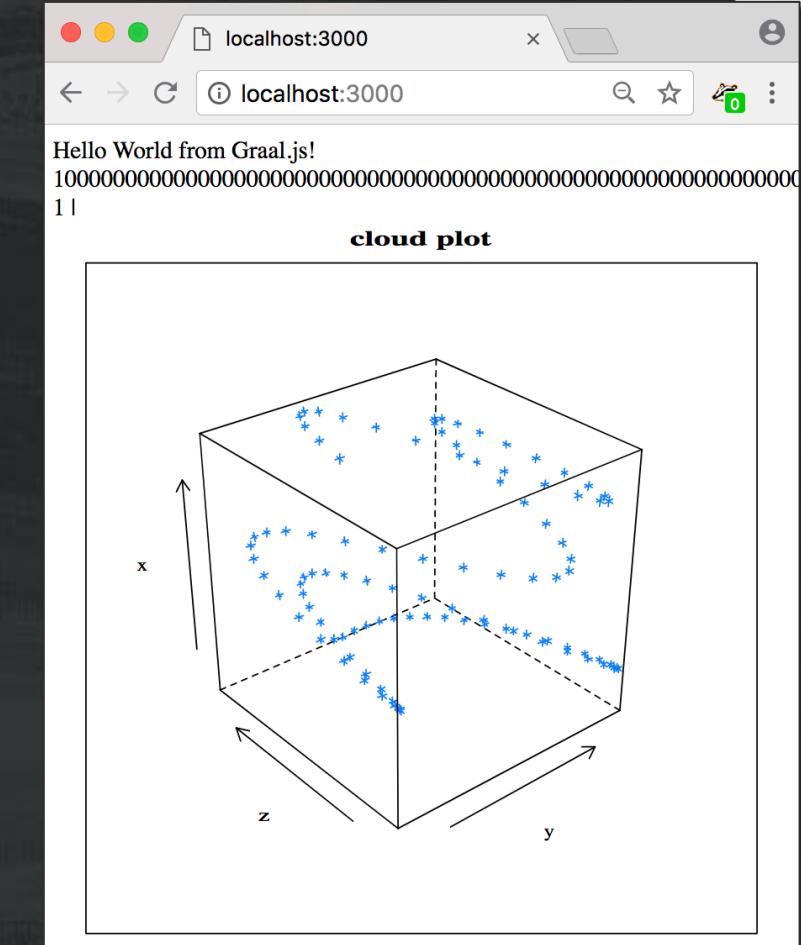
JavaScript/node.jsプログラムの中からJavaとR言語を呼ぶ例

```
const express = require('express')
const app = express()

const BigInteger = Java.type('java.math.BigInteger')

app.get('/', function (req, res) {
  var text = 'Hello World from Graal.js!<br>'
  // Using Java standard library classes
  text += BigInteger.valueOf(10).pow(100)
    .add(BigInteger.valueOf(1)).toString() + '<br>'
  // Using R interoperability to create graphs
  text += Polyglot.eval('R',
`svg();
require(lattice);
x <- 1:100
y <- sin(x/10)
z <- cos(x^1.3/(runif(1)*5+10))
print(cloud(x~y~z, main="cloud plot"))
grDevices:::svg.off()
`);
  res.send(text)
})
app.listen(3000, function () {
  console.log('Example app listening on port 3000!')
})
```





エンタープライズ分野への適用



既存Java
アプリケーション



ORACLE
WebLogic



マイクロサービスと
クラウド



ORACLE®
Cloud Infrastructure

MICRONAUT



多言語
アプリケーション



ハンズオン演習内容サマリー 1

■ 事前準備

- OCI上VMインスタンスを作成
- Oracle Linux 7.9, 4CPU, 32GBメモリ
- インスタンス作成時SSHキーを生成し、クライアントPCにダウンロード

■ 演習 1: GraalVMのインストール

- GraalVM Enterprise Edition 21.3 (LTS版) をインストール
- Native Image機能をインストール

■ 演習 2: 高性能JITコンパイラ

- Java Microbenmark Harness (JMH) を使用し、Stream APIを使用したJavaプログラムの実行時間を測定
- JITコンパイラ vs C2コンパイラ

ハンズオン演習内容サマリー 2

■ 演習 3: Native Image

- GraalVM Demos
- ディレクトリー配下すべてのファイルを回帰的に集計し、ファイルサイズの合計を集計するJavaプログラムを実行
- JITモード vs AOTモード（ネイティブモード）

■ 演習 4: GraalVMとJavaフレームワーク

- MicronautによるRESTfulサービスを作成し、起動時間を測定
- Spring BootによるRESTfulサービスを作成し、起動時間を測定
- JITモード vs AOTモード

■ 演習 5: GraalVMとDockerコンテナ

- Spring BootによるRESTfulサービスをコンテナにパッケージング
- 3種類のDockerコンテナを作成し、起動時間とコンテナサイズを比較
 - fat jar
 - native image
 - ほぼ静的なnative image

ハンズオンへのアクセス

- ハンズオンの前提条件
 - 1. 有効なOracleアカウント
 - Oracleアカウント（クラウドアカウントではありません）の作成方法
 - <https://www.oracle.com/jp/corporate/contact/contact-account-help/>
 - 上記ページ右上の「アカウントを表示」→「アカウントを作成」の順に作成頂けます。
 - 2. SSH接続可能なPC
 - 3. 安定したネットワーク環境

- ハンズオン・コンテンツへのアクセス
 - <https://apexapps.oracle.com/pls/apex/dbpm/r/livelabs/view-workshop?wid=3211>

ハンズオン実施環境の選択

- 「次世代Java高速実行基盤GraalVMハンズオン」のトップページにある「Run On Livelabs Sandbox」ボタンをクリックします。
- 事前作成済みのOracleアカウント情報を入力し、サインインします。
 - 未作成の場合「Create Account」ボタンをクリックして作成します。

[Japanese] Launch GraalVM to accelerate java workload and build cloud native application 次世代Java高速実行基盤GraalVMハンズオン Workshop

Get started with GraalVM and building cloud native application with high performance

Workshop length: 8 hours

Other LiveLabs you might like

Ways to run this workshop

Choose how you want to run this workshop.

Run On Your Tenancy

Using your credits | Services available | Free Tier

Run On LiveLabs Sandbox

You need an Oracle account to run on the free LiveLabs Sandbox: [Oracle account help](#) | [Oracle account signup](#)

Share Workshop Link

Workshop Details

Need help?

Oracle account sign in

Username

Password

Sign in

Need help?

Don't have an Oracle Account?

Create Account

© Oracle | Terms of Use | Privacy Policy

ハンズオン・コースの予約

- ハンズオンの予約画面にて、「I consent to receive reservation emails」をチェックし、「Submit Reservation」ボタンをクリックします。

Reserve [Japanese] Launch GraalVM to accelerate java workload 次世代Java高速実行基盤GraalVMハンズオン

About This Workshop

Get started with GraalVM and building cloud native application with high performance

Workshop Duration: 3 hours

Your email address
.com

Attendee Timezone
Japan (+09:00)

Start Workshop Now?

Choose a date to attend



Choose a time to attend



Watch this video to learn how to reserve a future reservation

All LiveLabs reservations are provided free of charge but are subject to Oracle's Terms of Use and are provided solely for the purpose of running the workshop specified for any given reservation.

Any breach of these terms may result in the immediate cancellation of this reservation and the deletion of all objects created as part of the reservation.

Objects created in LiveLabs workshop reservations only exist until the lease time expires.

By making this reservation you hereby agree to abide by these terms and conditions.

We need to send you emails directly related to this workshop reservation *

I consent to receive the reservation emails

Submit Reservation

Cancel

ハンズオン環境のプロビジョニング

- これによりクラウド上のハンズオン用環境のプロビジョニングが開始されます。次の画面より「View your reservation」をクリックして、プロビジョニング状況を確認します。
- 「Launch Workshop」ボタンが表示されたら、クリックしハンズオンを開始します。

Thank you for reserving the [Japanese] Launch GraalVM to accelerate java workload and build cloud native application 次世代Java高速実行基盤GraalVMハンズオン workshop!

We are preparing your reservation details. You will receive a confirmation email shortly. You can also login at anytime to check the status by clicking [here](#).

My Reservations

All your current workshop reservations are shown below. You can edit active or pending reservations, view workshop details, attend an available workshop, or delete a reservation.

Note: The status of your reservations will be emailed to you. Check your mail for any status updates.

Reservation Details	Status	Action
<p>[Japanese] Launch GraalVM to accelerate java workload and build cloud native application 次世代Java高速実行基盤GraalVMハンズオン</p> <p>Tuesday May 3rd, 10:28pm (22:28) US/Pacific</p>	Status: Available	Launch Workshop Details Remove

A blue arrow points from the 'View Your Reservations' button on the left to the 'Launch Workshop' button on the right.

ハンズオン注意事項

- VPNがSSH接続に影響を及ぼす場合、VPNを停止してください。
- SSHキーの格納場所と権限変更、および接続方法
 - Windows 10: PCログインユーザのディレクトリー配下（権限の変更は不要）
 - 例： `C:\Users\<ユーザ名>`
 - MacOS, Linux: 任意のディレクトリーに格納し、権限を以下のように変更
 - `chmod 600 <ダウンロードしたSSHキーファイル名>`
 - SSHキーによる接続方法
 - SSHキー格納場所よりOCIインスタンスのPublic IPに対してopcユーザ名で接続
 - `ssh -i <SSHキーファイル名> opc@<x.x.x.x>`
- SSH接続が中断された場合の対処
 - RESTfulサービス起動中にSSH接続が失われた場合
 - 上記手順でOCIインスタンスに対して再度SSH接続を行う
 - インスタンス上8080番ポートが開いている可能性がありますので、以下のコマンドで確認し、開いている場合は閉じてください。
 - `lsof -i -P | grep 8080` //プロセスIDを特定。存在する場合下記コマンドでプロセスを終了してください
 - `kill -9 <プロセスID>`
 - Dockerコンテナが起動中にSSH接続が失われた場合
 - 上記手順でOCIインスタンスに対して再度SSH接続を行う
 - インスタンス上コンテナが起動している可能性がありますので、以下のコマンドで確認、停止してください。
 - `sudo docker ps -a` //コンテナ状態を確認。起動中の場合下記コマンドを実行しコンテナを停止してください
 - `sudo docker rm -f `sudo docker ps -a -q``





Thank you