



# Oracle GraalVM

---

**Petr Novotny**

GraalVM Software Development Manager

Oracle Labs

11.11.2023

# Oracle Labs

Oracle Labs je výzkumná a vývojová organizace v rámci firmy Oracle.

V České republice má Oracle Labs týmy v Brně a v Praze.

Úzce spolupracujeme s VUT FIT a MatFyz UK formou studentských stáží a výzkumných grantů.

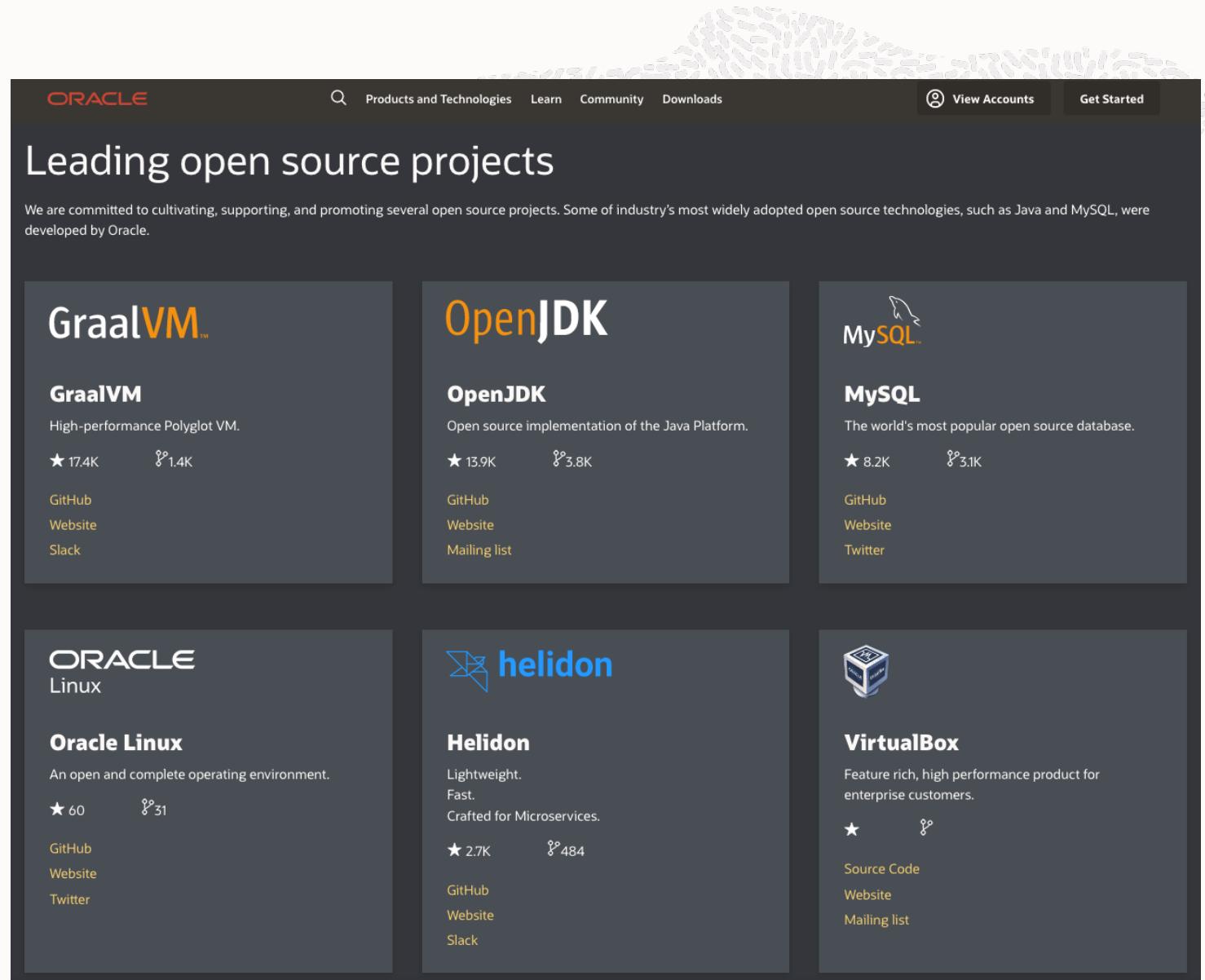


# Oracle Open Source

Oracle Labs iniciovaly a rozvíjejí hlavní open source projekty podporované firmou Oracle.



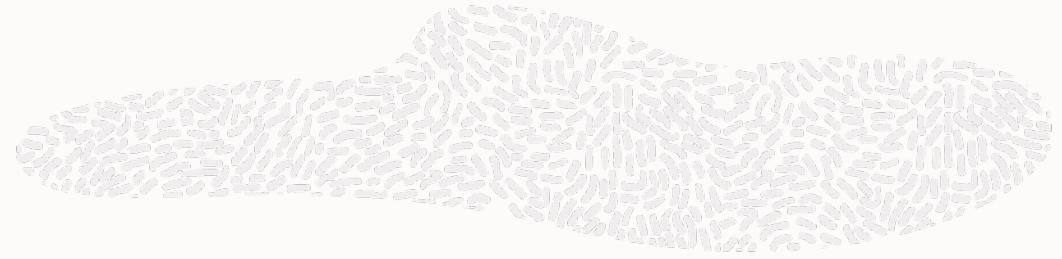
**HeatWave**



The screenshot shows the Oracle Open Source website with a dark header featuring the Oracle logo, a search bar, and navigation links for Products and Technologies, Learn, Community, Downloads, View Accounts, and Get Started. Below the header, a banner reads "Leading open source projects" with a subtext about Oracle's commitment to cultivating, supporting, and promoting open source projects. Six project cards are displayed in a grid:

- GraalVM**: High-performance Polyglot VM. (17.4K stars, 1.4K forks). Includes GitHub, Website, and Slack links.
- OpenJDK**: Open source implementation of the Java Platform. (13.9K stars, 3.8K forks). Includes GitHub, Website, and Mailing list links.
- MySQL**: The world's most popular open source database. (8.2K stars, 3.1K forks). Includes GitHub, Website, and Twitter links.
- Oracle Linux**: An open and complete operating environment. (60 stars, 31 forks). Includes GitHub, Website, and Twitter links.
- helidon**: Lightweight. Fast. Crafted for Microservices. (2.7K stars, 484 forks). Includes GitHub, Website, and Slack links.
- VirtualBox**: Feature rich, high performance product for enterprise customers. (No star or fork count shown). Includes Source Code, Website, and Mailing list links.

# GraalVM, Graal Cloud Native, GraalOS



Více informací na [graal.cloud](https://graal.cloud).

## What is the Graal Stack?

The Graal stack includes Graal Cloud Native, which makes it easy to build multicloud native applications; GraalVM Native Image, which compiles your application into an efficient native executable; and GraalOS, which makes the cloud as easy to use as the JVM.



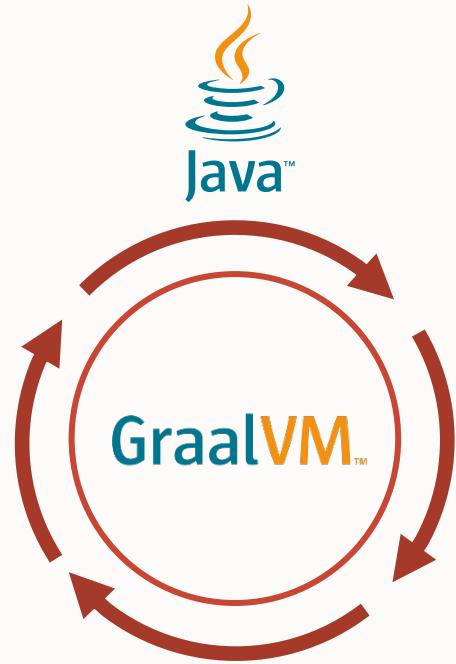
**Graal**VM™

**Graal** Cloud Native

**Graal**OS

# Co je GraalVM

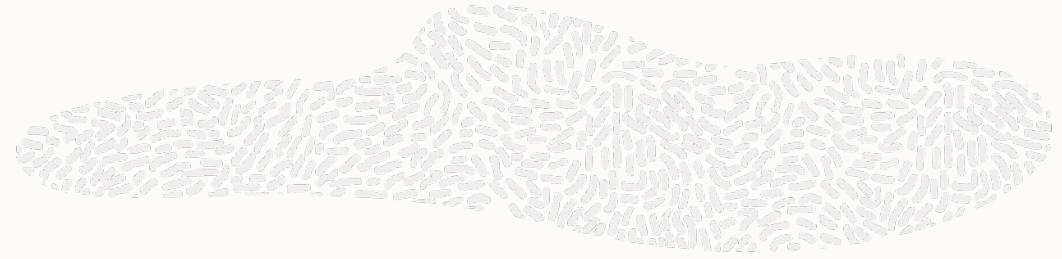
Více informací na [graalvm.org](https://graalvm.org).



Just-in-Time (JIT) kompilátor



Ahead-of-Time (AOT)  
kompilátor Native Image



GraalVM jako prostředí pro běh  
aplikací v různých jazycích

# Komunita kolem GraalVM

GraalVM má rozsáhlou a aktivní komunitu.

Spolupracujeme s dalšími open source projekty, např. v oblasti microservices jsou to projekty Spring, Quarkus, Micronaut a Helidon.

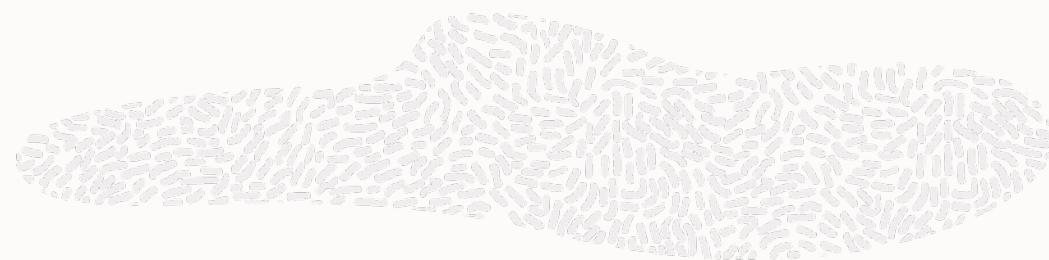




## **David Kozák**

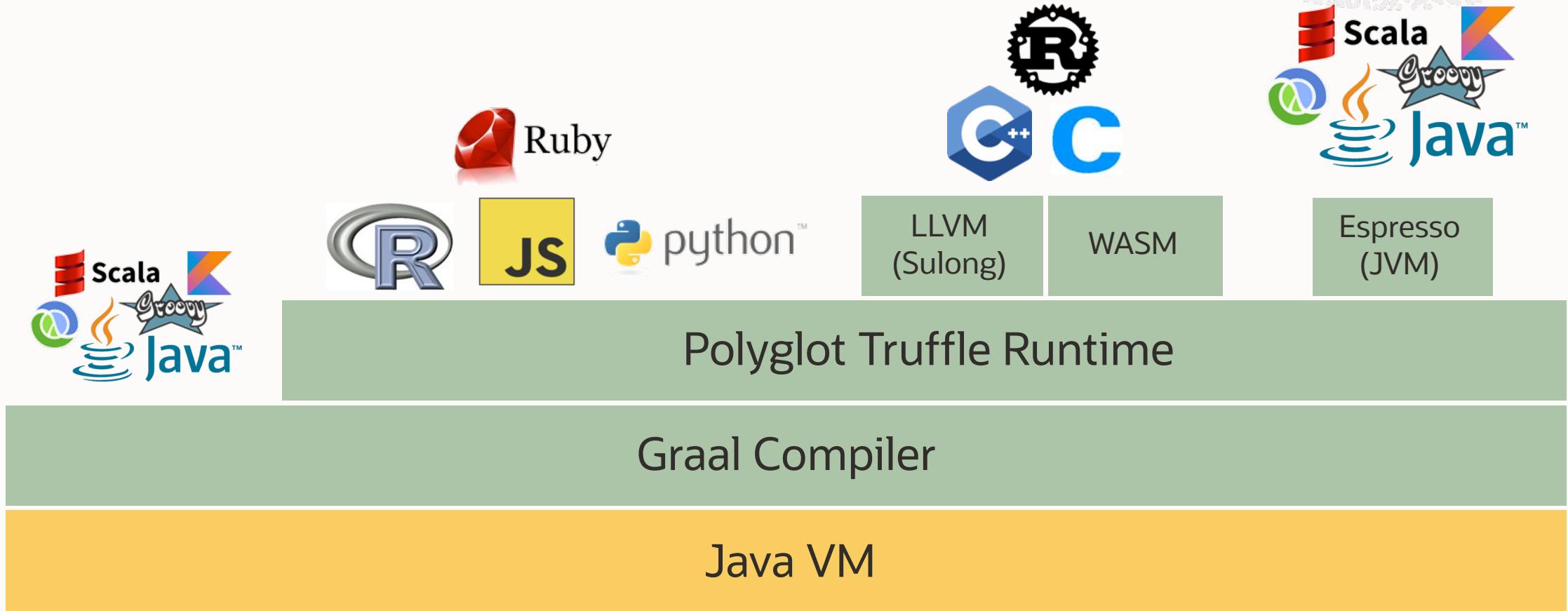
---

2020+ Research Assistant, Oracle Labs  
2021+ PhD student, FIT VUT



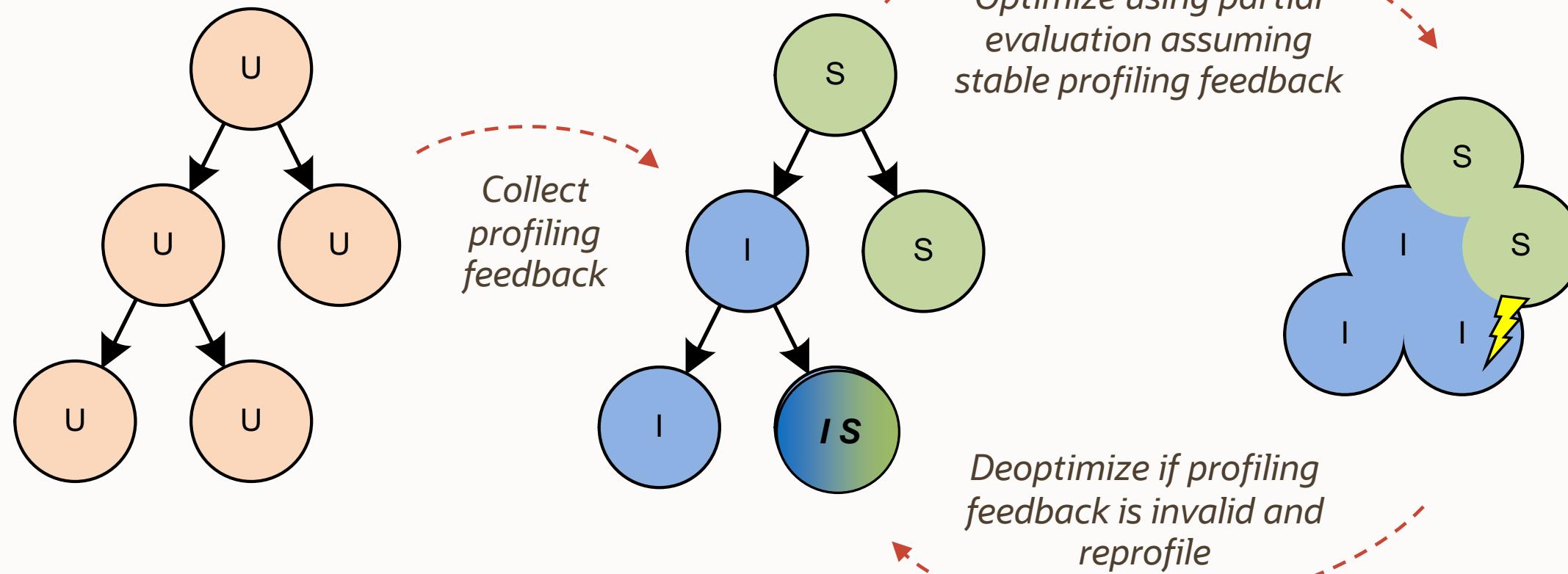
# GraalVM<sup>TM</sup>

**EXAMPLE**



# Dynamic speculation and deoptimization

One VM to Rule Them All  
Onward! 2013



# GraalVM v NetSuite

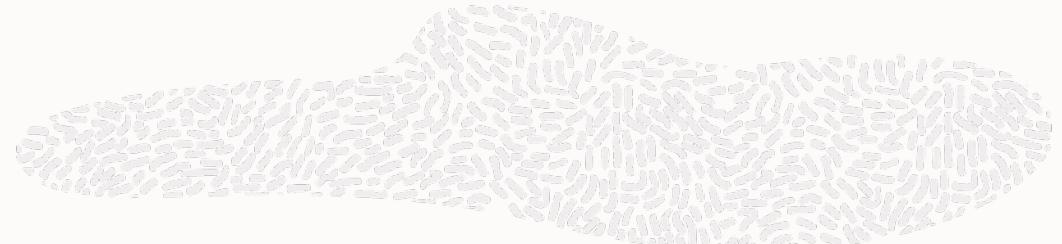
## NetSuite

- Integrovaný soubor cloud business aplikací.
- Rozsáhlá a komplexní Java aplikace.
- Nasazená na vysokém počtu serverů.

## SuiteScript

- Rozšíření JavaScriptu běžící na serveru.
- Přizpůsobení a automatizace business procesů.
- V současnosti JIT kompilován pomocí **GraalJS**.

Více informací v plánovaném blog postu.



**ORACLE®**  
**NETSUITE**

# From research to production

## MLE and the Future of Server-Side Programming in Oracle APEX

February 11, 2021 | 11 minute read



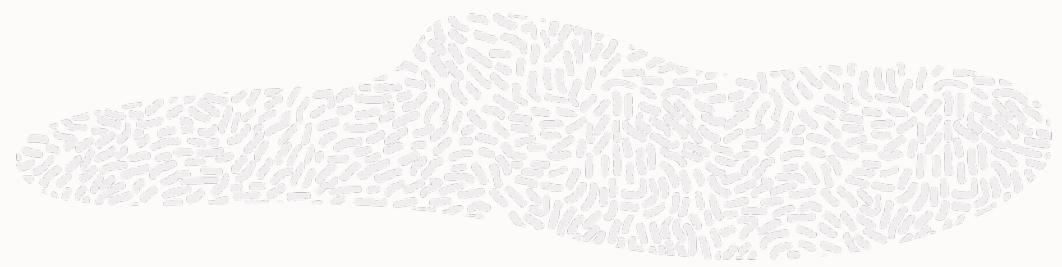
Salim Hlayel

Principal Product Manager



```
function extendProjectTasks( status ) {
    if (status != "Closed") {
        return true;
    }
    else {
        return false;
    }
}

for ( var row of apex.conn.execute( "select id,
    status from project_tasks where project = "
) ) {
```



# GraalVM Native Image

# EXAMPLE



# GraalVM™



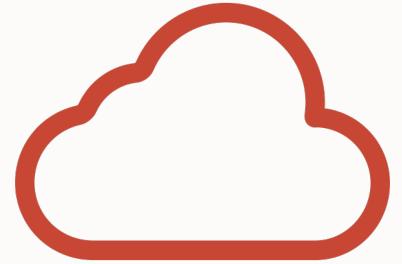
## JIT

```
java MyMainClass
```

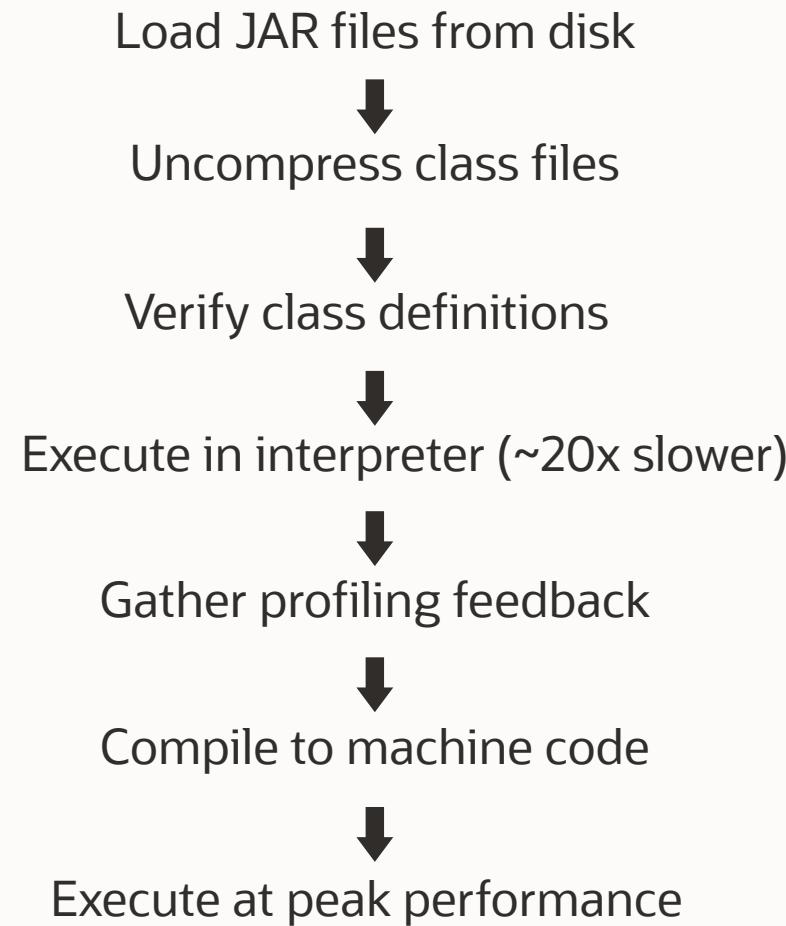


## AOT

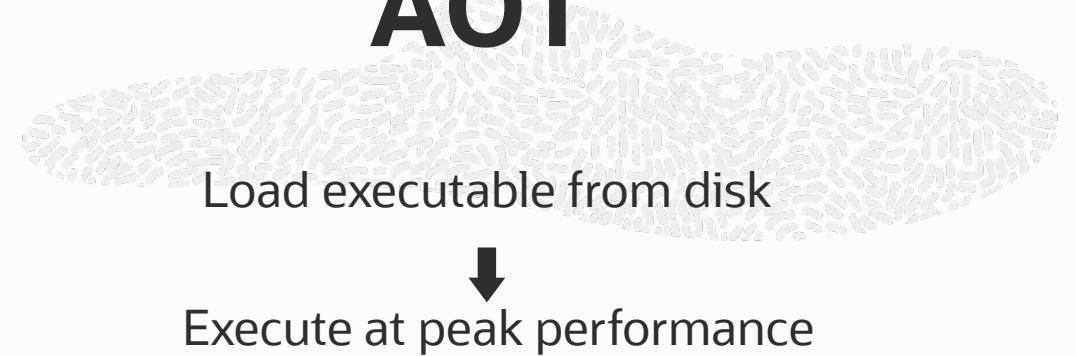
```
native-image MyMainClass  
./mymainclass
```



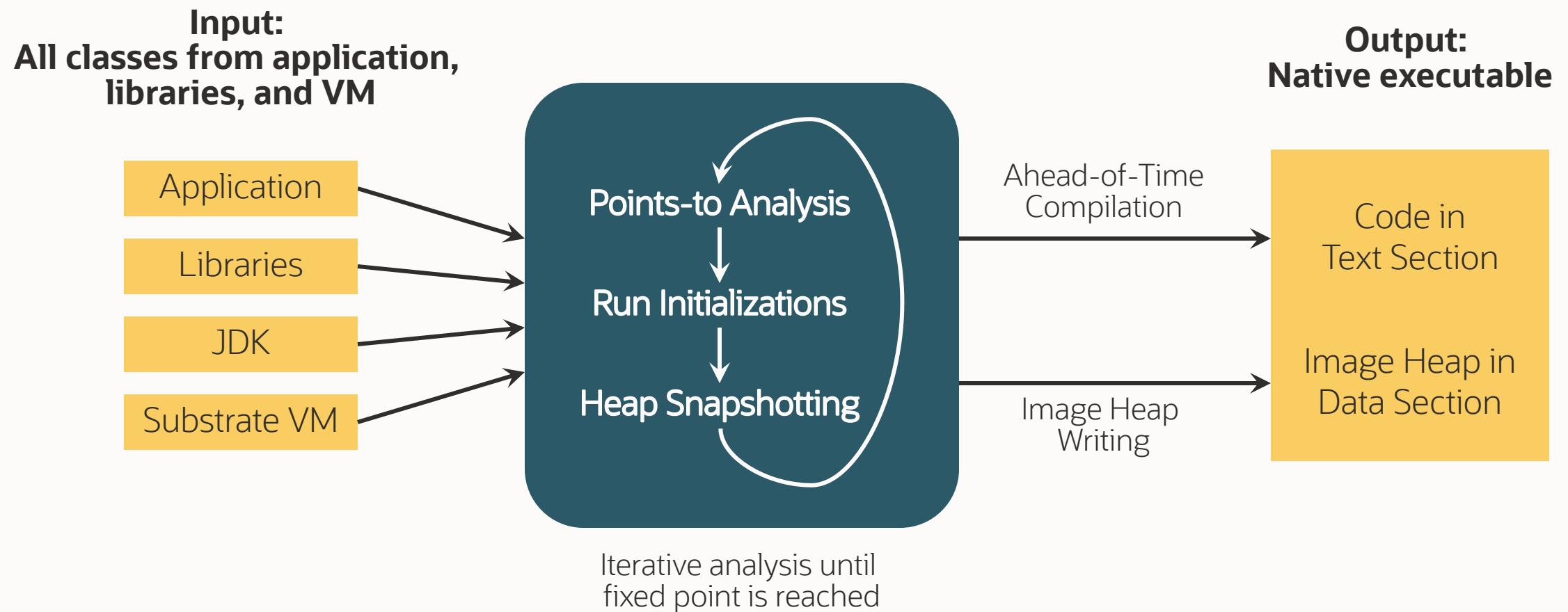
# JIT



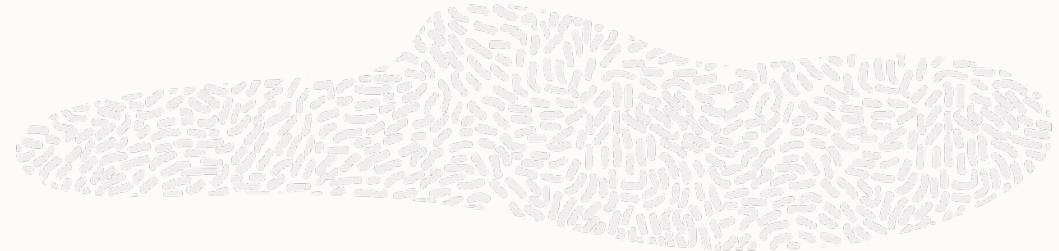
# AOT



# Native Image Build Process



# GraalVM & Reflection?



- GraalVM 🤝 Reflection!
- Native Image tries to resolve the target elements through a static analysis that detects calls to the Reflection API
  - If the analysis can not automatically detect your use of reflection, you might need additional configuration
- Trace reflection, JNI, resource usage on the JVM with the tracing agent
  - Manual adjustment / addition might still be necessary

The screenshot shows the GitHub repository page for 'oracle/graalvm-reachability-metadata'. The repository has 104 stars, 5 forks, and 171 commits. Recent activity includes a bump to version 0.2.3-SNAPSHOT by dnestoro. The repository description states: 'Repository which contains community-driven collection of GraalVM reachability metadata for open-source libraries.' It includes links to the README, CC0-1.0 license, code of conduct, and statistics like 104 stars and 9 forks.

Code

Issues 9 Pull requests 1 Discussions Actions Projects Wiki Security Insights Settings

master 6 branches 6 tags

Go to file Add file Code About

dnestoro Bump repo version to 0.2.3-SNAPSHOT 02da975 23 hours ago 171 commits

.github Bump actions/setup-graalvm version 18 days ago

docs Fix JSON quotation marks 4 months ago

gradle Relaxed checkstyle requirements. 2 months ago

metadata Consul api 1.4.5 (#1) 6 days ago

tests Consul api 1.4.5 (#1) 6 days ago

.gitignore Simplify .gitignore 5 months ago

.gitmodules Add graalvm/setup-graalvm as a submodule 3 months ago

CONTRIBUTING.md Enable override attribute on Netty dependencies 23 days ago

Releases 6

Release 0.2.2 Latest 23 hours ago

# Java in the Cloud - Goals



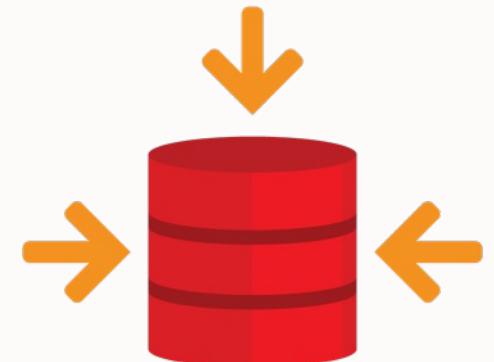
**Start Fast**



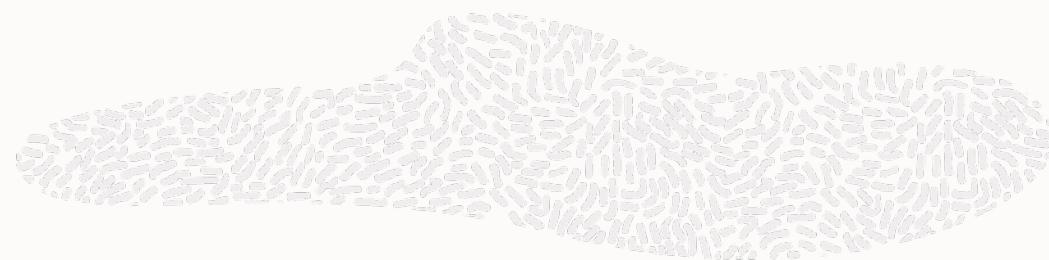
**Low Resource Usage**



**Minimize Vulnerability**

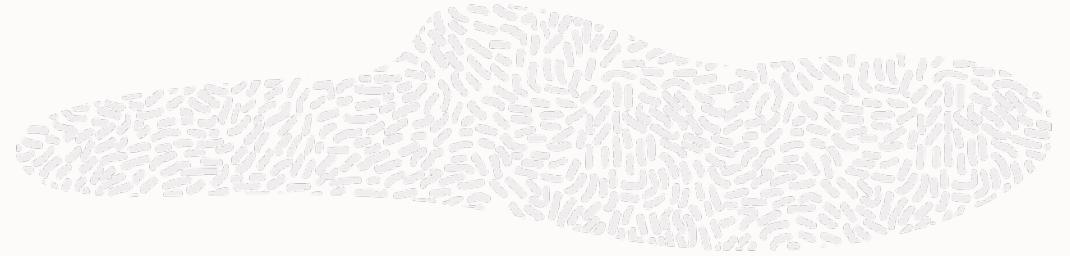


**Compact Packaging**



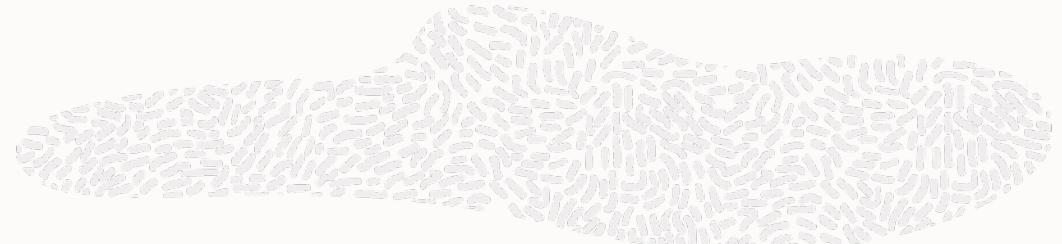
# Vývoj Native Image

Mé zkušenosti z dlouhodobé stáže



# První větší úkol - Lokalizace

# Lokalizace v Javě



Řešena pomocí **Resource Bundles**:

- Mapování klíč => hodnota s jednoduchým konceptem dědičnosti.

```
abstract class ResourceBundle {  
    /** Get the string value for a given key */  
    public String getString(String key);  
    /** Get the object for a given key */  
    public Object getObject(String key);  
}
```



class PropertyResourceBundle

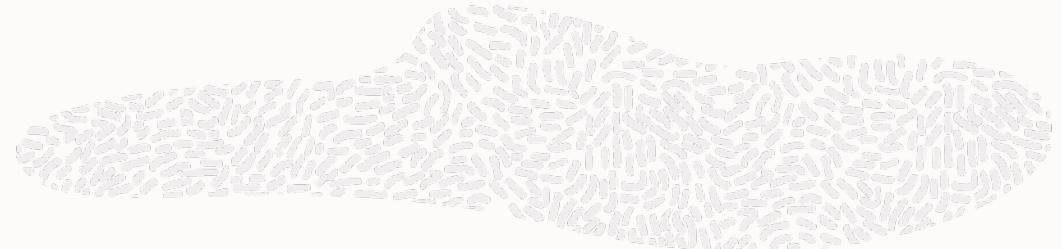


```
abstract class ListResourceBundle {  
    abstract Object[][] getContents();  
}
```



**EXAMPLE**

# Lokalizace v Javě



## Jak identifikujeme správnou Resource Bundle?

```
public static ResourceBundle getBundle(String baseName, Locale targetLocale,  
          ClassLoader loader, Control control)
```

- Pomerně komplexní implementace.
- Využívá reflexi. 

# Lokalizace v Native Image



1. Všechny **ResourceBundles** dostupné za běhu musí být inicializovány při překladu.
2. Objekty uloženy v **mapě**.
3. Kód  **ResourceBundle.getBundle()** **substituován**.

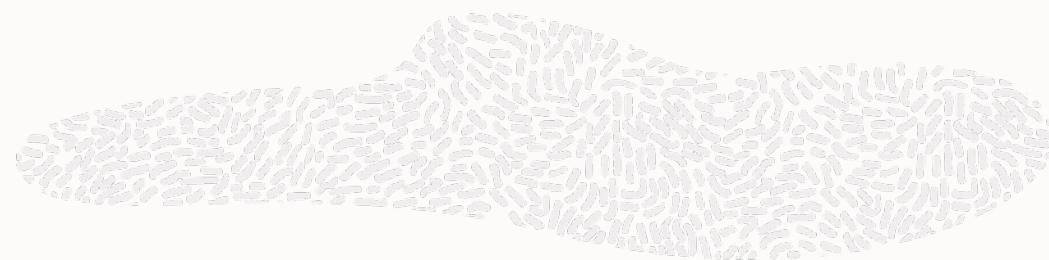
```
@TargetClass(java.util.ResourceBundle.class)
class Target_java_util_ResourceBundle {
    @Substitute
    ResourceBundle getBundle(...) {...}
}
```

# Problémy lokalizace



- Substituování `ResourceBundle.getBundle()` může být zdrojem nekonzistence mezi JVM a Native Image.
  - Implementujeme vyhledávání opravdu totožným způsobem?
- Ne všechny **ResourceBundles** mohou být inicializovány v době překladu.
  - Riziko selhání překladu s kryptickým chybovým hlášením.

**“Můžeme využít původní JDK kód místo substituce?”**



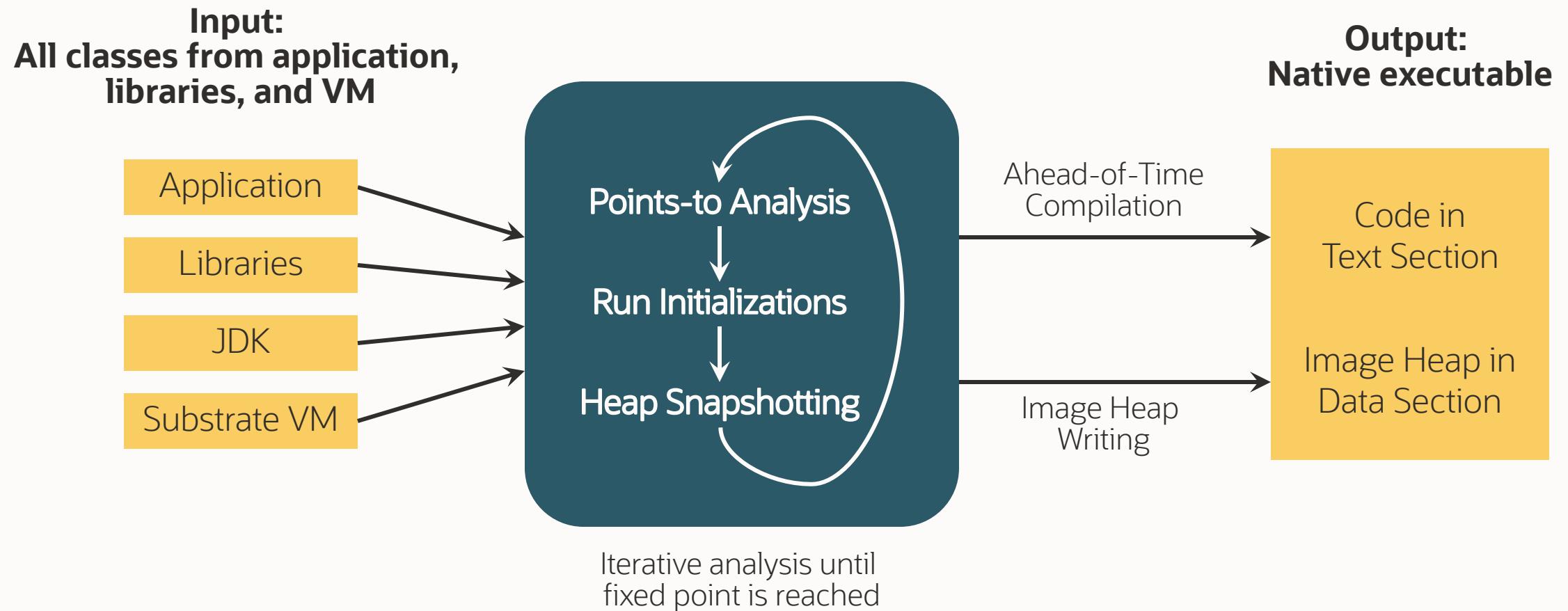
## Jak jsem vyrobil 300MB HelloWorld...

```
class FormatData_af extends ListResourceBundle {  
    Object[][] getContents() {...}  
}
```

```
@TargetClass(java.util.ListResourceBundle.class)  
class Target_java_util_ListResourceBundle {  
    @Substitute  
    private void loadLookup() {...}  
}
```

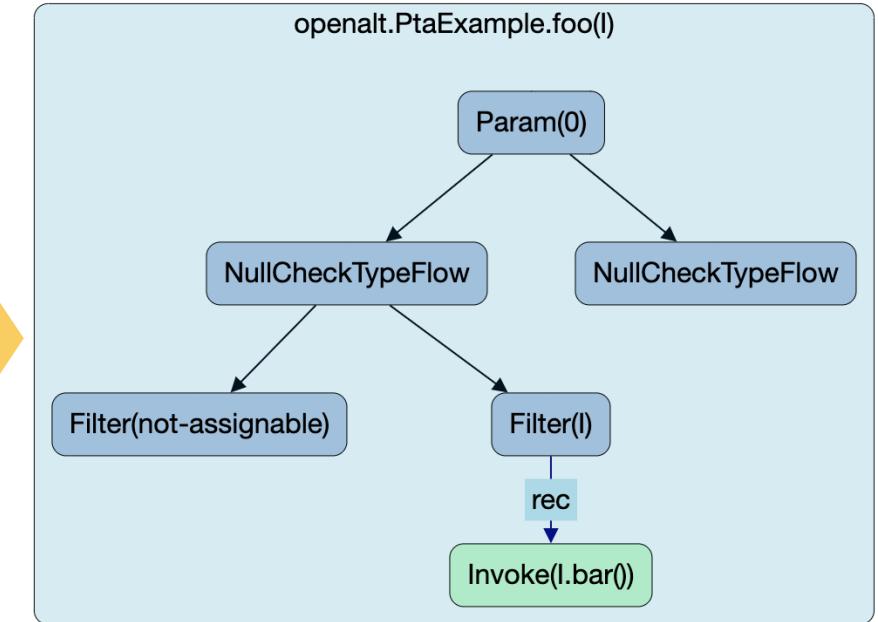
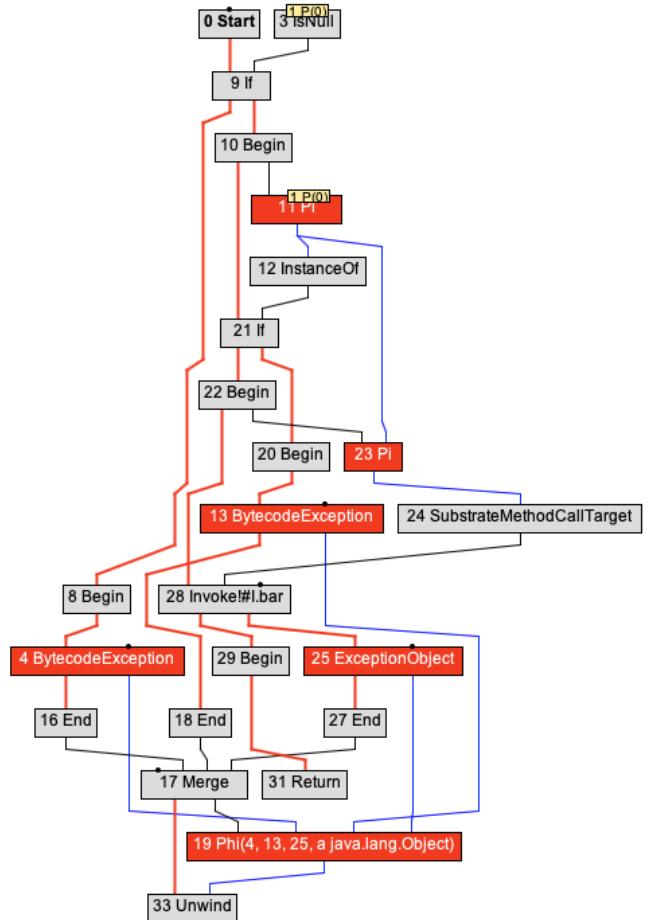
EXAMPLE

# Native Image Build Process

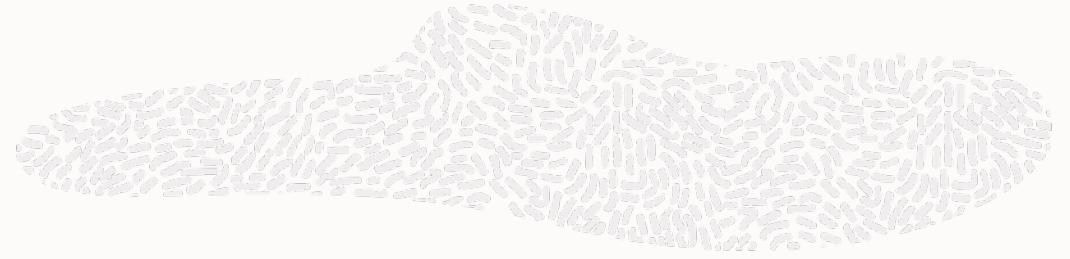


# Points-to analýza pod pokličkou

```
void foo(I i) {  
    i.bar();  
}
```



**Analýza Spring Petclinic trvá přes 90 vteřin...**



# Můžeme použít jinou metodu?



# Comparing Rapid Type Analysis with Points-To Analysis in GraalVM Native Image

David Kozak

ikozak@fit.vut.cz

Brno University of Technology  
Czechia

Vojin Jovanovic

vojin.jovanovic@oracle.com

Oracle Labs  
Switzerland

Codrut Stancu

codrut.stancu@oracle.com

Oracle Labs  
Switzerland

Tomáš Vojnar

vojnar@fit.vut.cz

Brno University of Technology  
Czechia

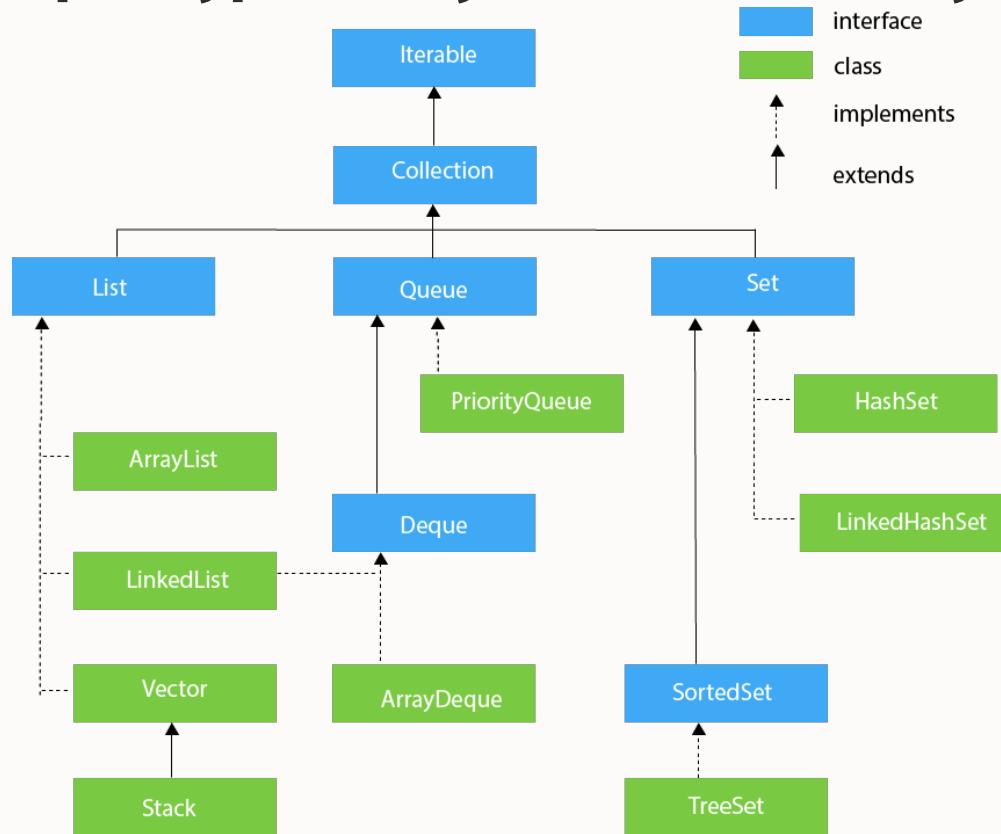
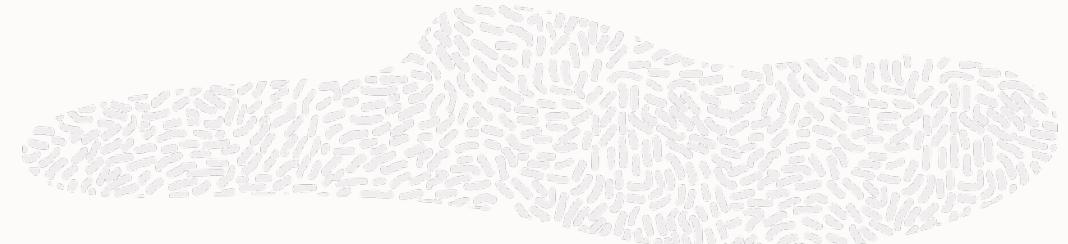
Christian Wimmer

christian.wimmer@oracle.com

Oracle Labs  
USA

<https://dl.acm.org/doi/abs/10.1145/3617651.3622980>

# Rapid type analýza – základní myšlenka



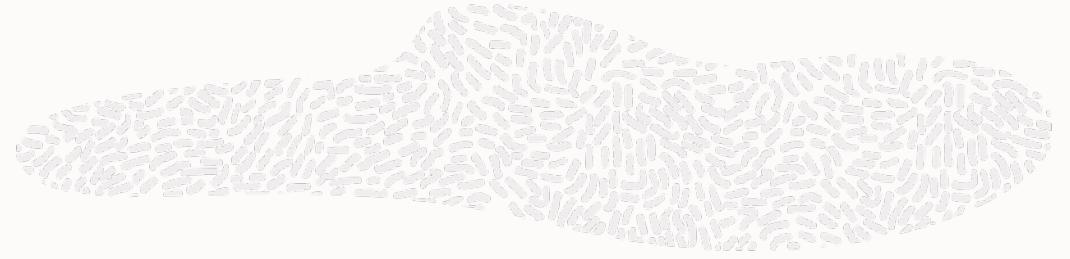
```
List<Object> list = new ArrayList();
```

## Typová hierarchie + Instanciované typy

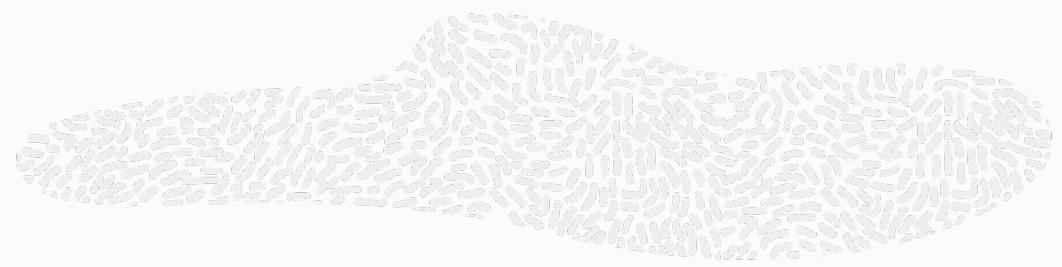


Suite	Benchmark	Reachable Methods PTA	Reachable Methods RTA	Analysis Time (s) PTA	Analysis Time (s) RTA	Total time (s) PTA	Total time (s) RTA	Binary size (MB) PTA	Binary size (MB) RTA
Console	helloworld	18	+17%	14	+21%	36	+17%	13	+23%
Dacapo	avrora	24	+25%	12	-8%	51	+6%	23	+30%
	fop	94	+4%	46	-30%	128	-10%	105	+11%
	jython	71	+8%	55	-35%	140	-26%	134	+9%
	luindex	26	+23%	13	-8%	54	+7%	32	+25%
Microservices	micronaut-helloworld-wrk	74	+4%	34	-32%	88	-9%	45	+18%
	mushop:order	168	+2%	102	-59%	209	-30%	104	+13%
	mushop:payment	82	+4%	36	-33%	91	-10%	50	+14%
	mushop:user	115	+3%	57	-44%	135	-18%	76	+13%
	petclinic-wrk	207	+4%	159	-64%	297	-35%	144	+15%
	quarkus-helloworld-wrk	52	+6%	18	-22%	69	-3%	50	+4%
	quarkus:registry	111	+5%	49	-39%	126	-16%	69	+19%
	spring-helloworld-wrk	67	+4%	30	-33%	87	-10%	47	+13%
	tika-wrk	82	+6%	29	-28%	117	-6%	88	+6%
	chi-square	173	+8%	129	-60%	260	-30%	100	+17%
Renaissance	dec-tree	324	+6%	2009	-95%	X	X	X	X
	future-genetic	27	+22%	15	0%	44	+5%	19	+21%
	gauss-mix	189	+8%	146	-61%	286	-32%	107	+17%
	log-regression	334	+7%	2215	-95%	X	X	X	X
	page-rank	171	+8%	129	-60%	258	-31%	119	+13%
	reactors	30	+13%	19	+16%	47	+11%	19	+21%
	scala-stm-bench7	30	+20%	19	+26%	49	+14%	19	+21%





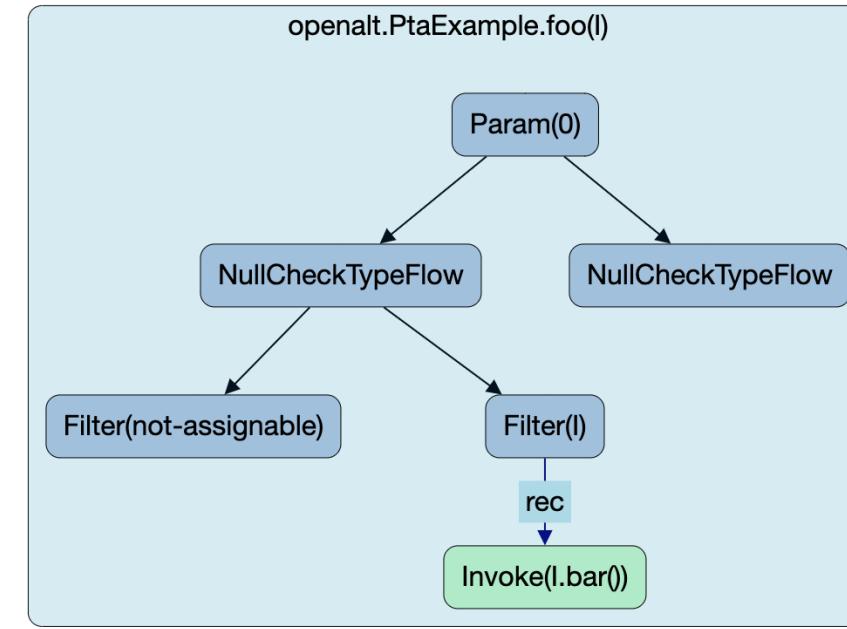
# Super, vyhrála RTA?

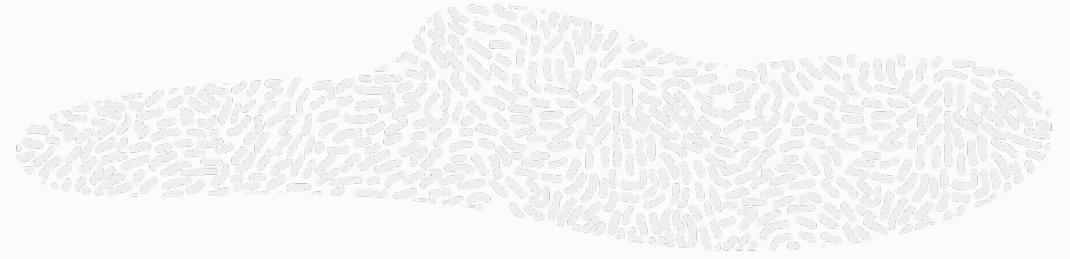


# Ne tak docela...

## Záleží na tom, kolik typů je v našem invoke?

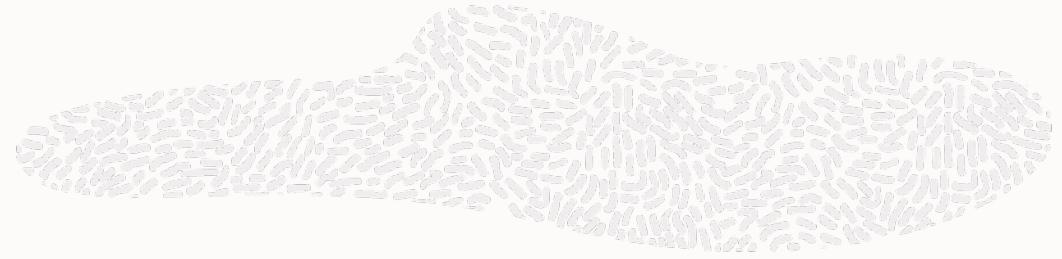
- Malý počet je pro překladač zajímavý.
- 20 vs 25 už tolik ne...





**If you can't beat them....  
...join them!**

# Závěr



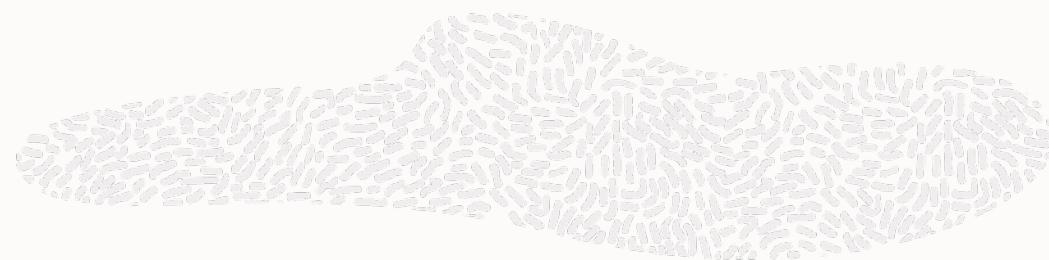
- GraalVM
- Native Image
- Zkušenosti ze stáže v Native Image
  - GraalVM je velký projekt – **možnosti stáže jsou velmi rozmanité**
- Možnosti do budoucna
  - Specializované prezentace
  - GraalVM, Truffle, GraalJS, GraalOS, Graal Could Next, Micronaut

# Děkujeme za pozornost

---

## Q & A



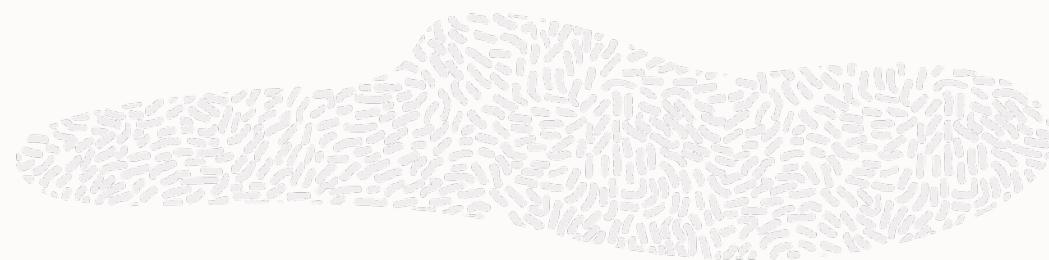


# Jak jsem se ke GraalVM dostal?



+



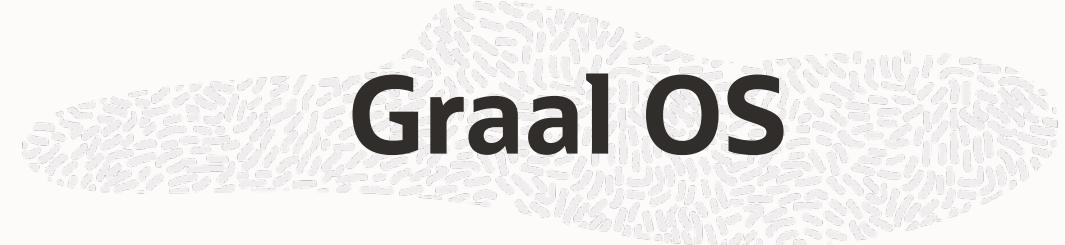


# ORACLE<sup>®</sup>

# NETSUITE

# Graal Cloud Native

“Build portable cloud native Java microservices that start instantly and use fewer resources to reduce compute costs.”



# Graal OS

“High-performance serverless application deployment platform.”

<https://graal.cloud/>