GraalVM™

Run Programs Faster Anywhere

WHY GRAALVM

GET STARTED

Issues

- '***Java is slow***'
- 'Write once, run everywhere'
- Different tools for different environments

High Java memory usage even for small programs

21

4

I have a couple of simple applications written in java, one of them written to act as a widget. What surprised me how much RAM even small applications use.

I wrote the following to see if it is a bug in my programs, or a general Java issue:

```
public class ram {
    public static void main(String[] args){
    while(true)System.out.print("Hello World");//while loop to give me time to check RAM usage
    }
}
```

Then compiled and ran it with `java ram` and it gave me the following RAM usage:

```
The process java (with pid 4489) is using approximately 43.3 MB of memory.
34460 KB    [heap]
7088 KB /usr/lib/jvm/java-7-openjdk/jre/lib/amd64/server/libjvm.so
1712 KB /usr/lib/jvm/java-7-openjdk/jre/lib/rt.jar
136 KB  [stack:4495]
120 KB  /usr/lib/jvm/java-7-openjdk/jre/lib/amd64/libjava.so
```

Isn't this too high? Especially a heap of 34MB. My system is ArchLinux x86_64 and openjdk-7.

Is there any way to minimise the amount of RAM used by the JVM?

Edit: I tried using the -Xmx flag and this is what I got (1281k was the smallest it would let me start with):

```
java -Xmx1281k ram
The process java (with pid 4987) is using approximately 27.6 MB of memory.
18388 KB    [heap]
```

For comparison, Python2 uses 4.4MB, Mono uses 4.3MB.

java memory ram

share edit flag

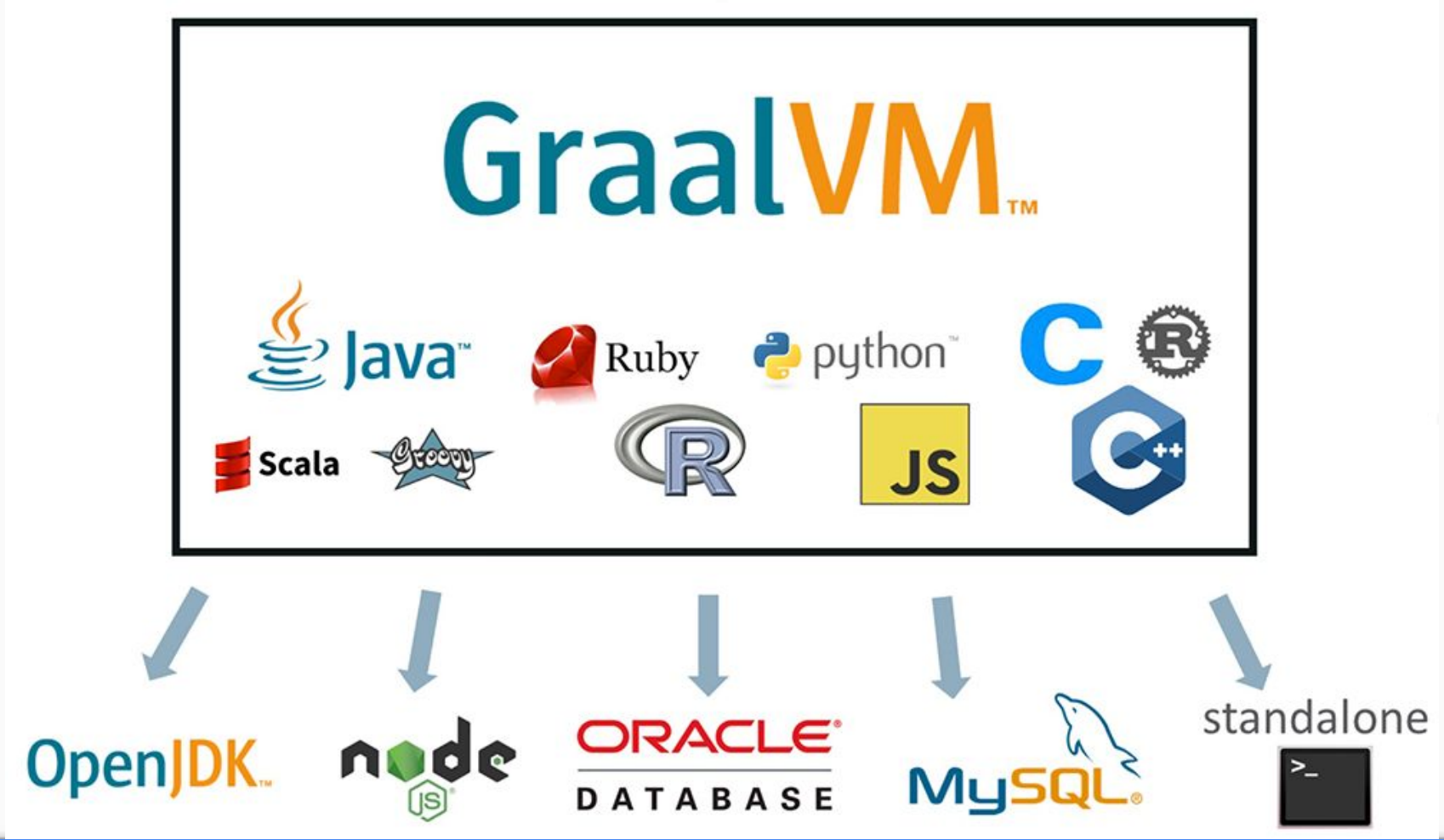
edited Dec 4 '12 at 0:15

asked Dec 3 '12 at 21:39

Vivo

106 ●1 ●1 ●5

https://stackoverflow.com/questions/13692206/high-java-memory-usage-even-for-small-programs



http://bigdatapassion.pl/blog/java/java-how-it-works/

GraalVM - what it gives

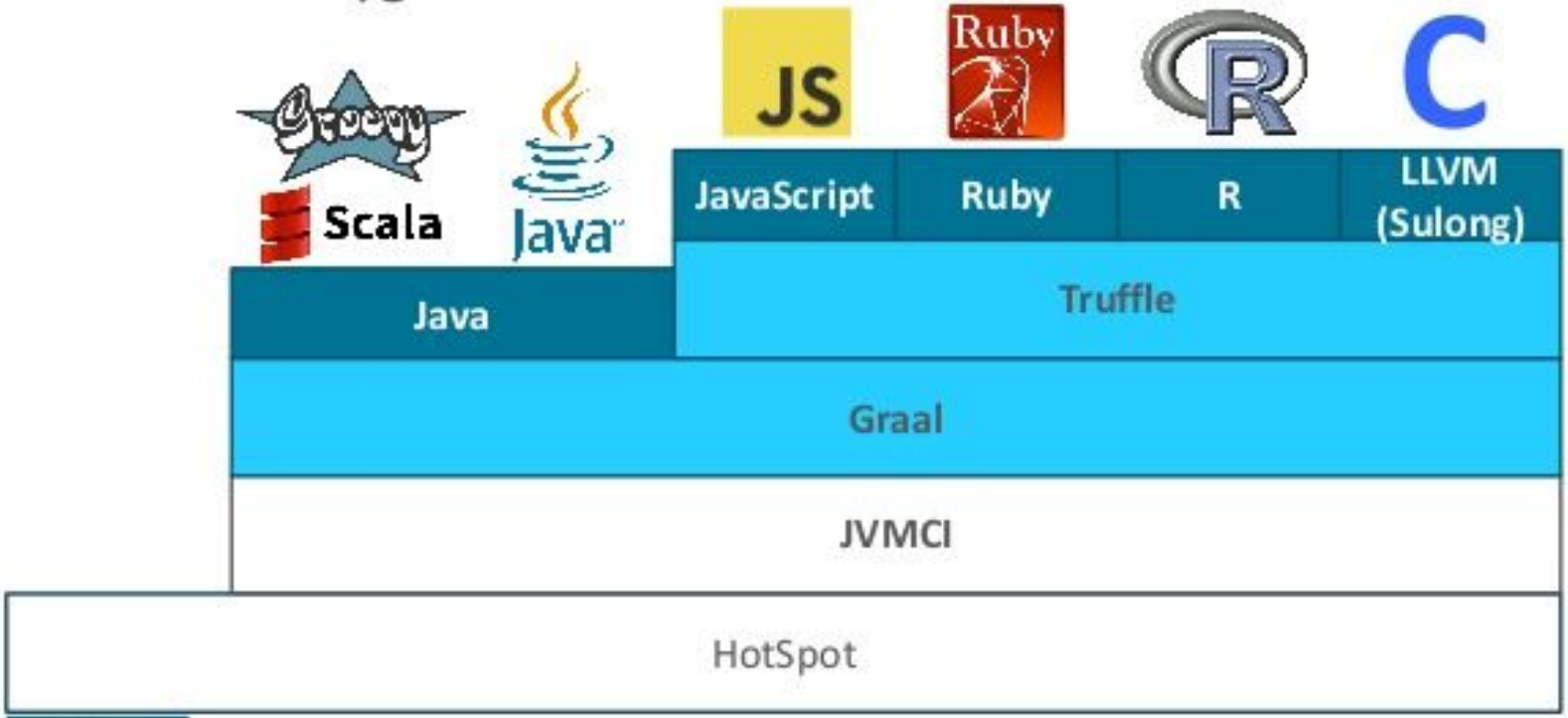
- Performance out-of-the-box
- Language interoperability
- Native Images

FFAQ

- ***Can I run it on windows?*** Under development, RC version available (for 6+ months?)
- ***Does it support Java 11+?*** Yes...and No. https://github.com/oracle/graal/issues/651
- ***How to use it?*** Just replace OpenJDK with it and run
- ***Who does use it?*** ie. Twitter. On production https://www.forbes.com/sites/oracle/2018/04/18/graalvm-1-0-gives-developers-a-speedy-polyglot-runtime-and-helps-twitter-save-money/#16b27a423936
- ***Is it free?*** Both Community and Enterprise versions are available
- ***Is it production ready?*** … Rather experimental - a lot may change, but yes
- ***Does it support reflection?*** Yes*
- ***What languages are supported now?*** JVM based, JavaScript, Ruby, R, Python*, LLVM based
- ***How?*** Truffle + Java compiler + Optimizations like partial escape analysis (https://www.beyondjava.net/escape-analysis-java), object flattening etc (https://medium.com/graalvm/under-the-hood-of-graalvm-jit-optimizations-d6e931394797)
- ...

Polyglot applications & tools

Graal VM Polyglot



http://gigasquidsoftware.com/blog/2017/10/22/embedded-interop-between-clojure-r-and-python-with-graalvm/

#VDZ19

```
(master+)$ cat hello-js.sql
DELIMITER $$
CREATE FUNCTION hello_js_world() RETURNS VARCHAR(255) LANGUAGE JS [[ function() {
  return "Hello, JS World!";
}]]$$
DELIMITER ;

select ROUTINE_NAME, EXTERNAL_LANGUAGE from information_schema.routines where ROUTINE_SCHEMA="demo";

SELECT hello_js_world() as Greeting;
(master+)$ mysql
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 36
Server version: 8.0.12 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

VOXXED DAYS ZÜRICH

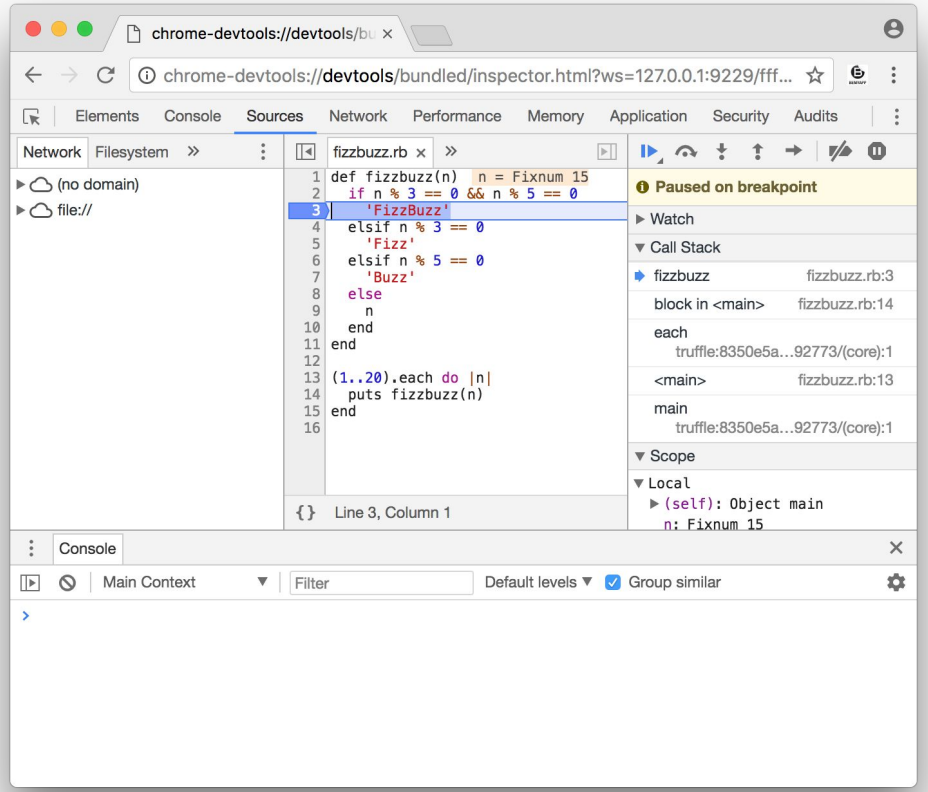
11:38 / 20:10

ORACLE

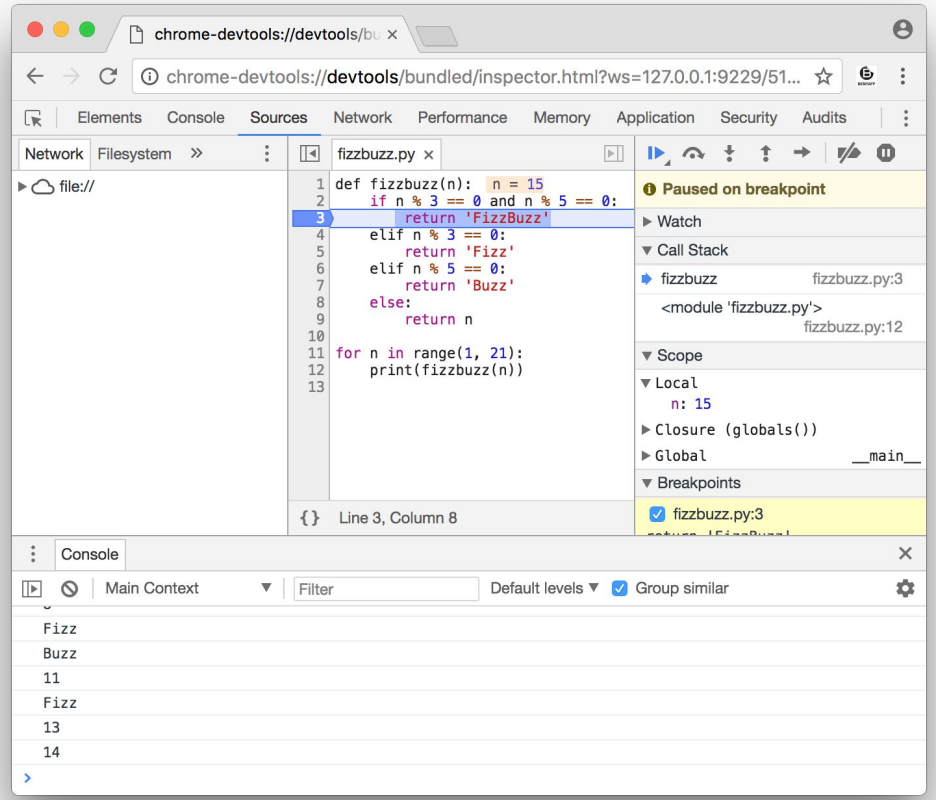
SWISS POST

https://www.youtube.com/watch?v=fAuTMvpsvRw

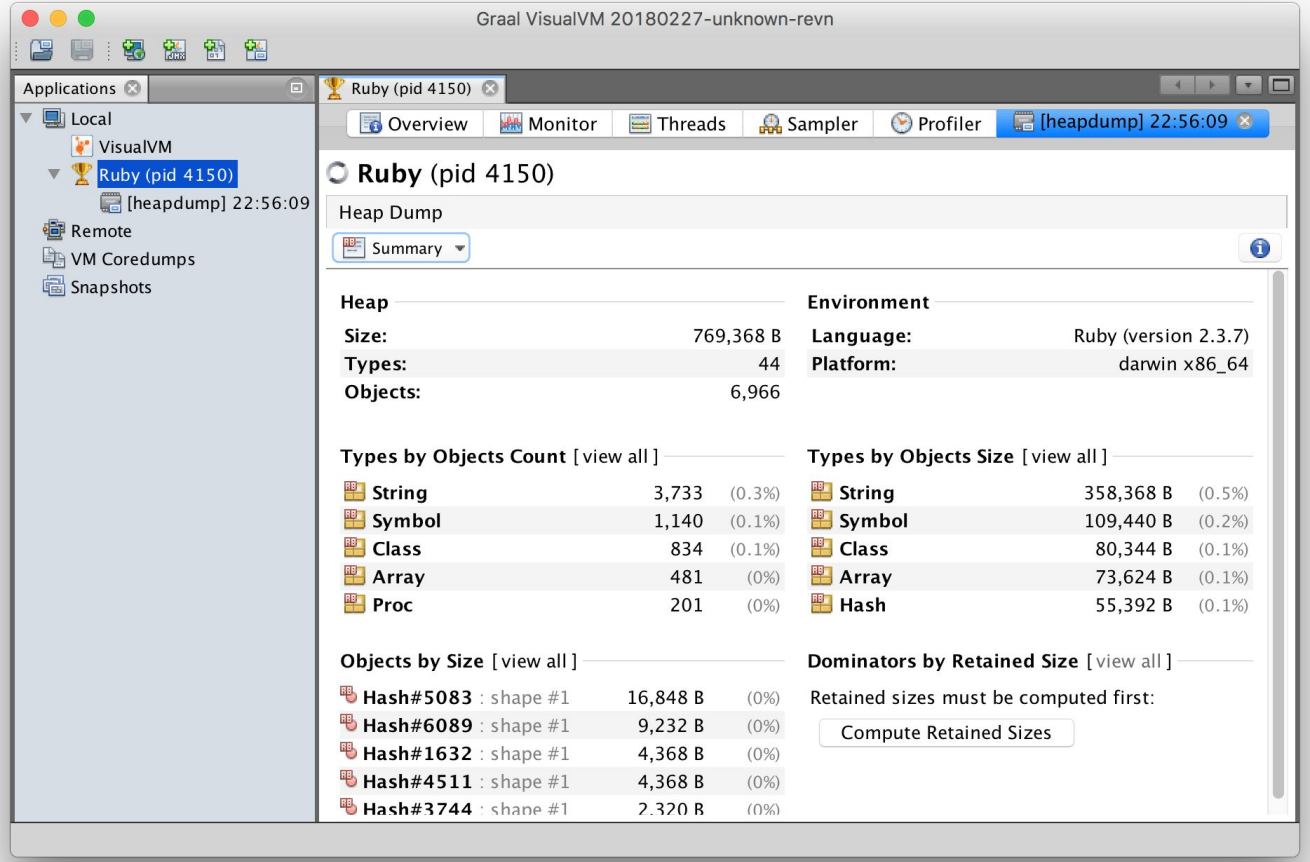
Ruby code



Python code



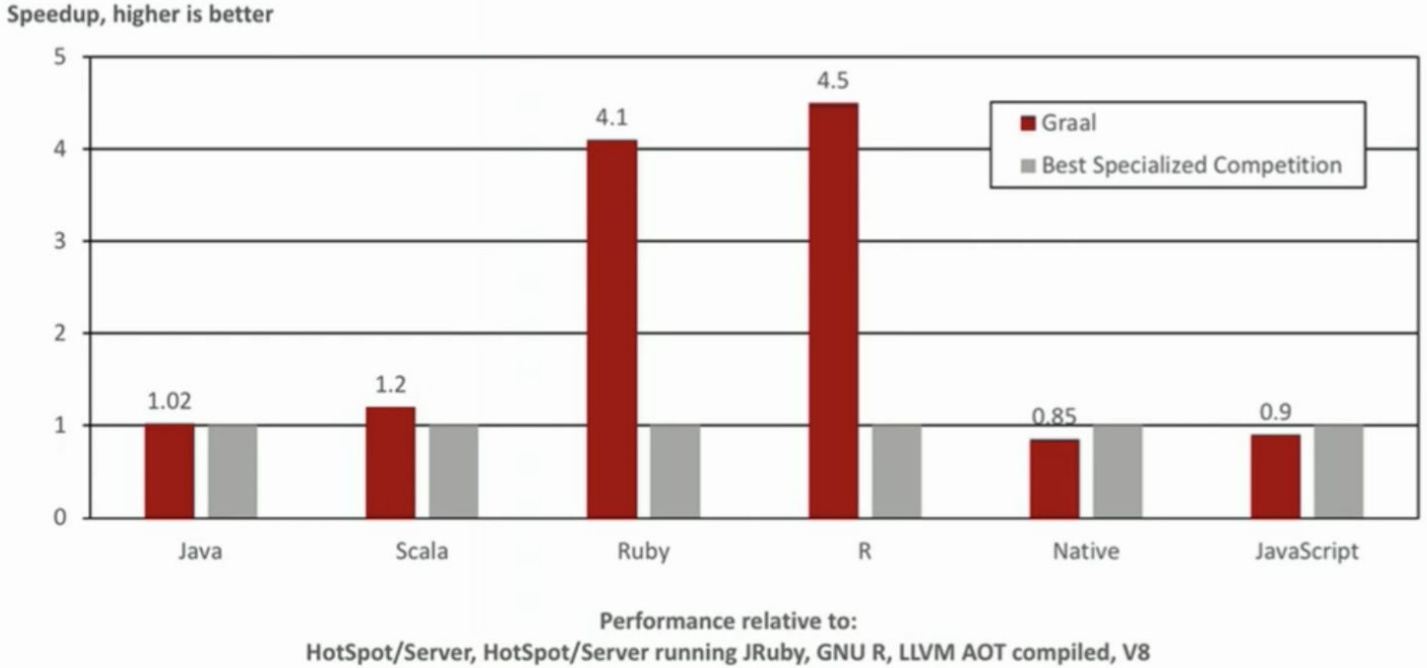
https://medium.com/graalvm/graalvm-ten-things-12d9111f307d



https://medium.com/graalvm/graalvm-ten-things-12d9111f307d

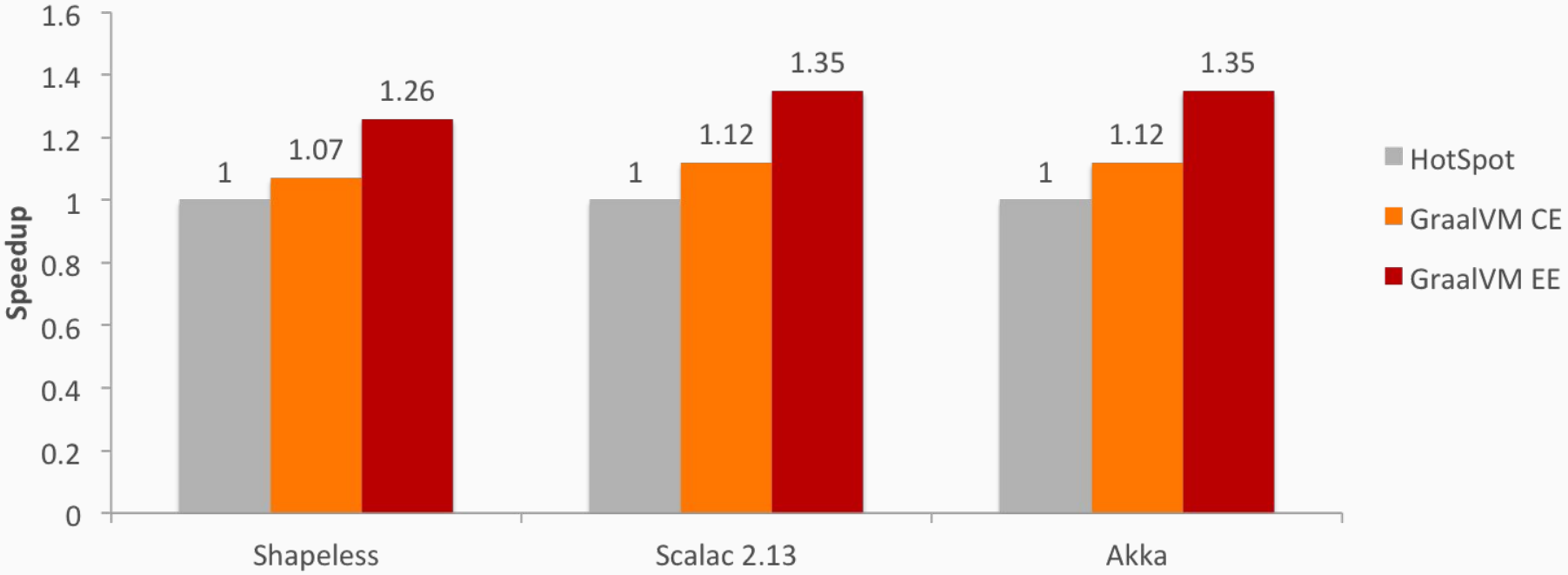
Performance

Performance: Graal VM

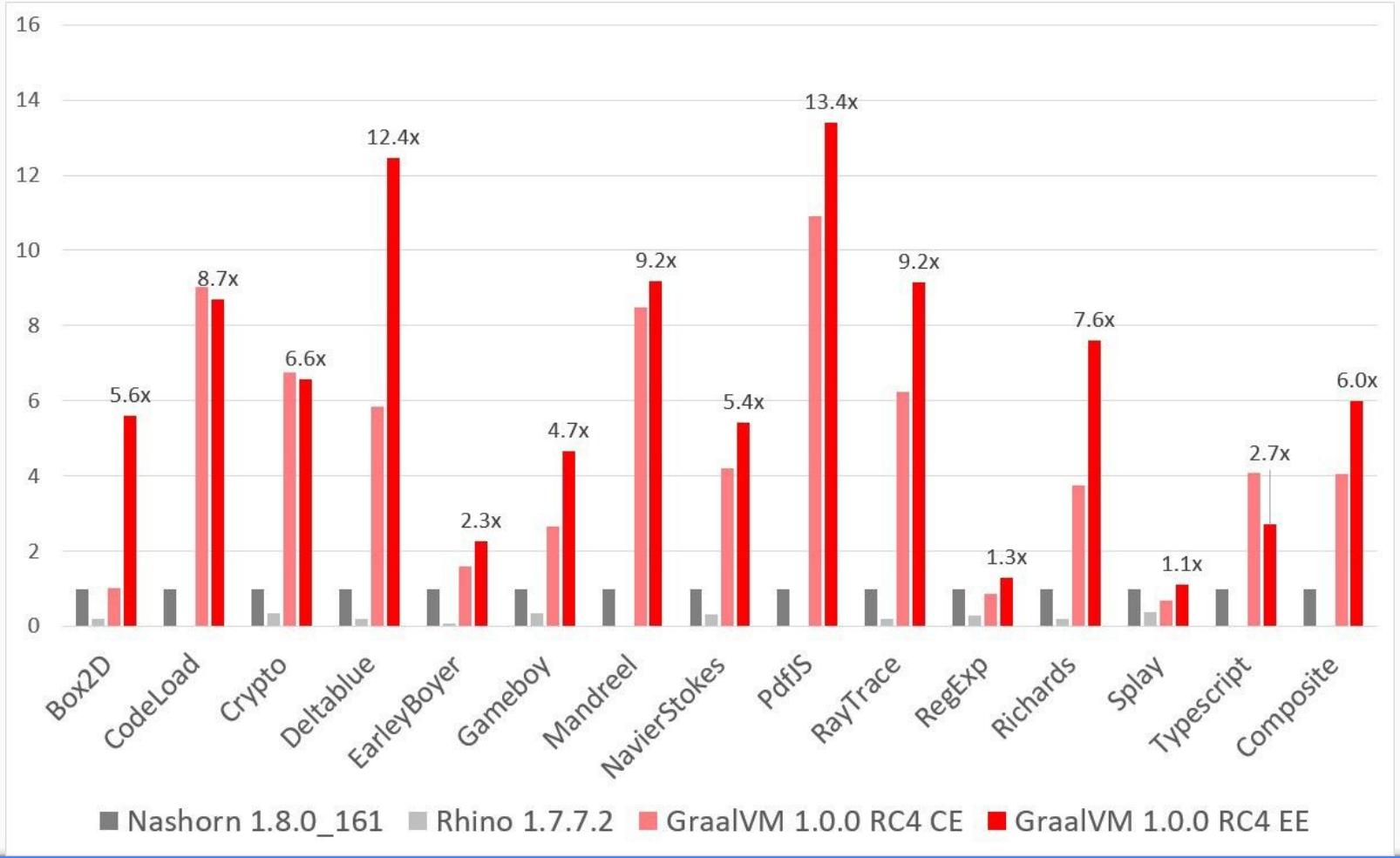


ORACLE

https://hackernoon.com/why-the-java-community-should-embrace-graalvm-abd3ea9121b5



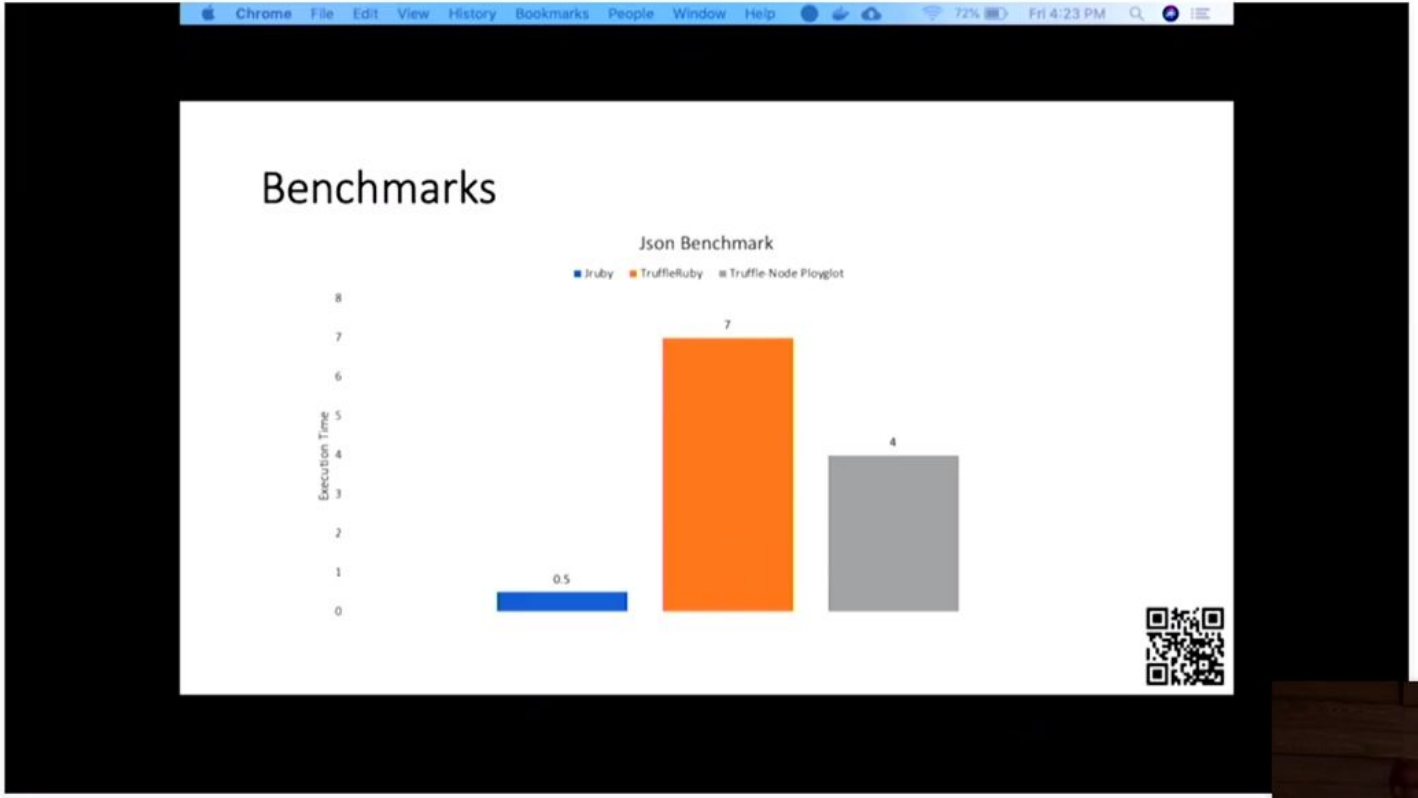
https://medium.com/graalvm/compiling-scala-faster-with-graalvm-86c5c0857fa3



https://medium.com/graalvm/oracle-graalvm-announces-support-for-nashorn-migration-c04810d75c1f

	96%	98%	98%	97%	7%	28%	97%
Feature name ▶	Edge 17	FF 61	CH 67, OP 54	Node >=8.10 <9[3]	JJS 1.8	JJS 10	GraalVM 1.0[4]
Syntax							
default function parameters	7/7	7/7	7/7	7/7	0/7	4/7	7/7
rest parameters	5/5	5/5	5/5	5/5	0/5	0/5	5/5
spread (...) operator	15/15	15/15	15/15	15/15	0/15	0/15	15/15
object literal extensions	6/6	6/6	6/6	6/6	0/6	2/6	6/6
for..of loops	9/9	9/9	9/9	9/9	0/9	4/9	9/9
octal and binary literals	4/4	4/4	4/4	4/4	0/4	2/4	4/4
template literals	5/5	5/5	5/5	5/5	0/5	3/5	5/5
RegExp "y" and "u" flags	5/5	5/5	5/5	5/5	0/5	0/5	5/5
destructuring, declarations	22/22	22/22	22/22	22/22	0/22	0/22	22/22
destructuring, assignment	24/24	24/24	24/24	24/24	0/24	0/24	24/24
destructuring, parameters	23/24	24/24	24/24	24/24	0/24	0/24	24/24
Unicode code point escapes	2/2	2/2	2/2	2/2	0/2	0/2	2/2
new.target	2/2	2/2	2/2	2/2	0/2	0/2	2/2

https://medium.com/graalvm/oracle-graalvm-announces-support-for-nashorn-migration-c04810d75c1f



foss asia

OpenTech SUMMIT

Lifelong Learning Institute
Singapore
14-17 March, 2019

17:49 / 47:54

https://www.youtube.com/watch?v=mJcMM3wZA20

Native Images

Native images

Pre-requirements

- JEP 317: Experimental Java-Based JIT Compiler - OpenJDK - Java 10
- JEP 243: Java-Level JVM Compiler Interface - OpenJDK - Java 10
- JEP 295: Ahead-of-Time Compilation - Java 9

Pros

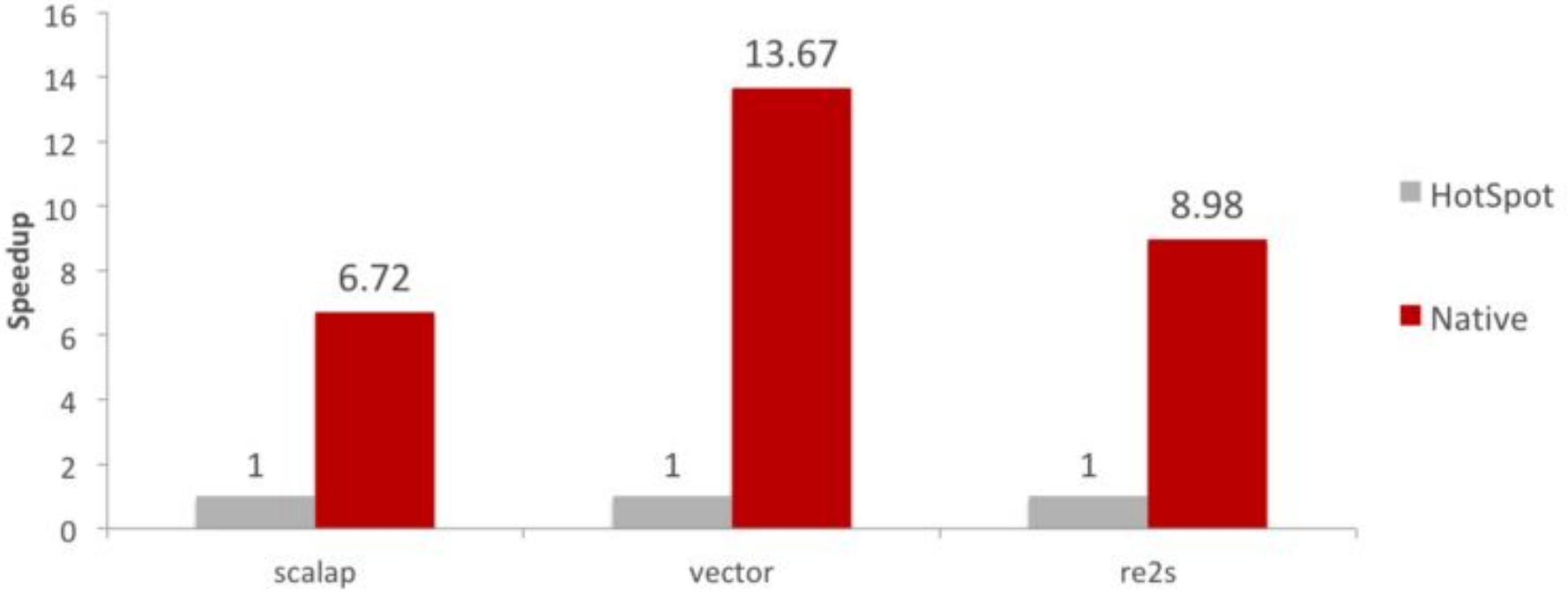
- Lower memory footprint
- Out of HotSpot (Runs on SubstrateVM)
- MUCH Faster startup
- Secured execution (native LLVM apps)

```
Erans-MBP:graalvm-1.0.0-rc1 eranlevy$ Contents/Home/bin/./native-image --language:js SumNum
Build on Server(pid: 11044, port: 26681)
   classlist:   1,147.71 ms
       (cap):   5,262.46 ms
       setup:   6,676.23 ms
  (typeflow):  18,093.06 ms
   (objects):  17,226.01 ms
  (features):   2,816.21 ms
    analysis:  39,556.06 ms
6572 method(s) included for runtime compilation
     universe:   1,906.22 ms
      (parse):   5,630.36 ms
     (inline):   3,382.46 ms
    (compile):  32,248.81 ms
      compile:  43,914.26 ms
        image:   6,821.54 ms
        write:  35,143.02 ms
      [total]: 135,274.83 ms
Erans-MBP:graalvm-1.0.0-rc1 eranlevy$ ls
Contents        SumNum.class    SumNum.java     sumnum          test.json
Erans-MBP:graalvm-1.0.0-rc1 eranlevy$ ./sumnum
3
Erans-MBP:graalvm-1.0.0-rc1 eranlevy$
```

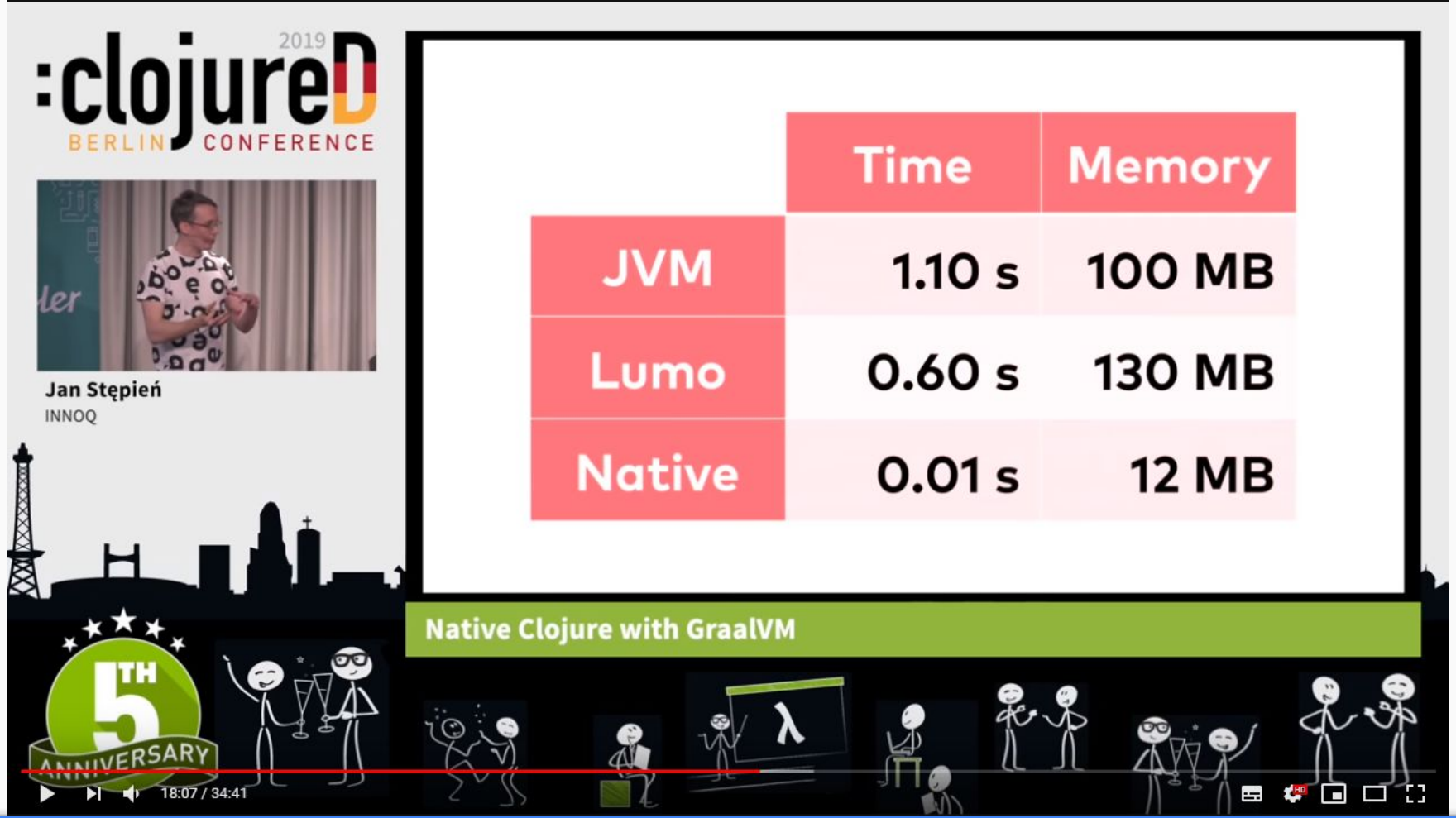
By now only JVM languages can be compiled to native images.
However, polyglot JVM applications can be compiled to native, so through this - other also.
https://www.graalvm.org/docs/reference-manual/aot-compilation/

There should be Spring kofu native image demo, but…
https://github.com/spring-projects/spring-fu/issues/198

Scala app startup time

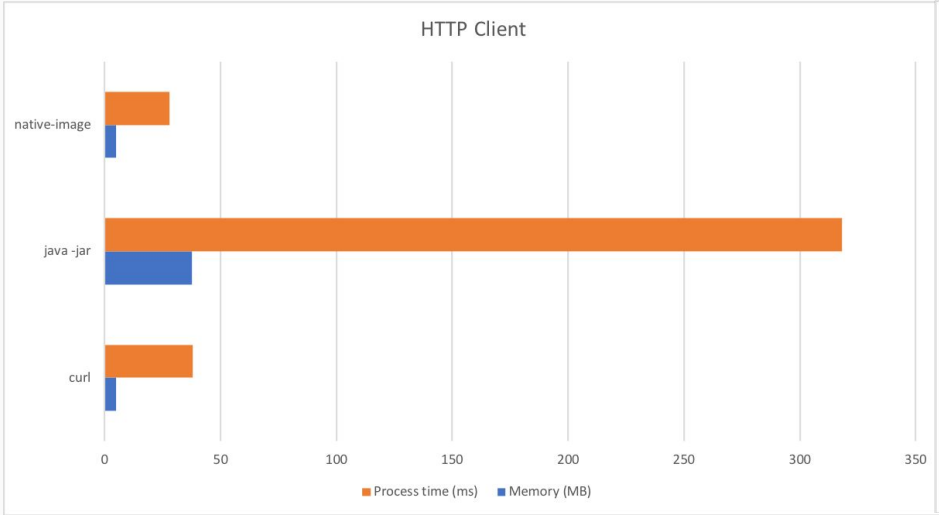


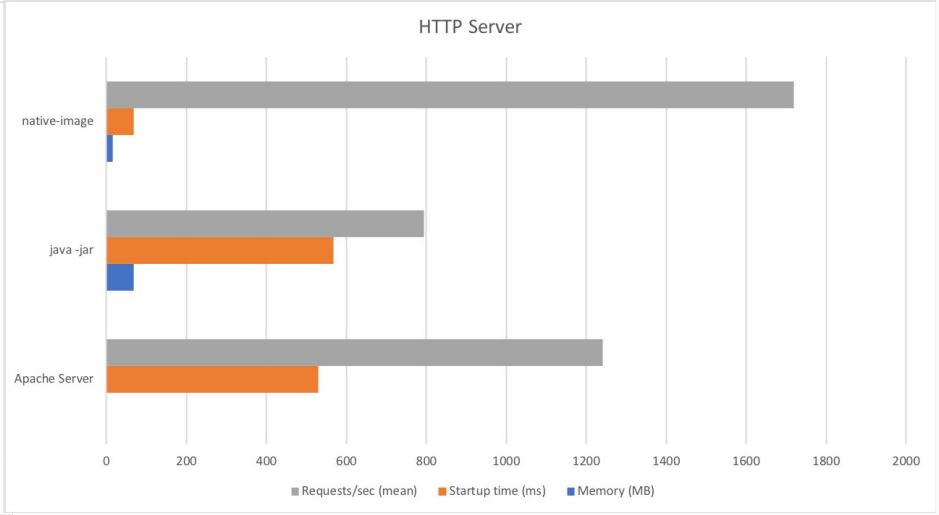
https://medium.com/graalvm/compiling-scala-faster-with-graalvm-86c5c0857fa3



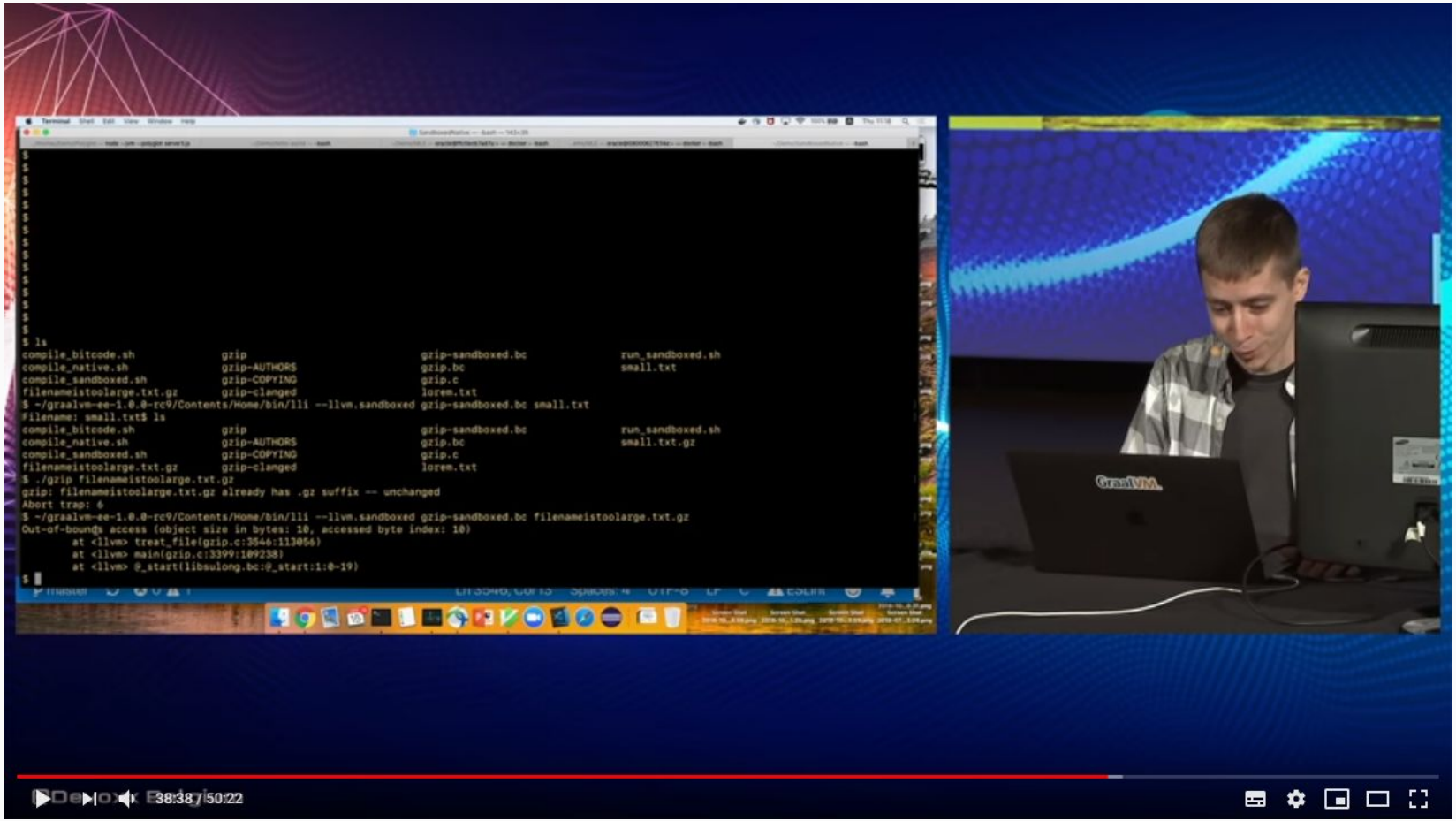
https://www.youtube.com/watch?v=topKYJgv6qA

Case study: Java HTTP client/server





https://sites.google.com/a/athaydes.com/renato-athaydes/posts/a7mbnative-imagejavaappthatrunsin30msandusesonly4mbofram



https://www.youtube.com/watch?v=50JxcnvJjMQ

Resources

- https://github.com/graalvm/examples - examples
- https://github.com/oracle/graal - graal repository
- https://medium.com/graalvm - medium blog
- https://medium.com/graalvm/graalvm-ten-things-12d9111f307d nice introduction
- https://www.youtube.com/watch?v=a-XEZobXspo - slides on Devoxx 3h
- https://www.youtube.com/watch?v=oN3QOsZ1KAw - spring boot & GraalVM native image
- https://www.youtube.com/watch?v=GmxXgUkOOdw - Js on GraalVM
- https://www.youtube.com/watch?v=50JxcnvJjMQ - Q&A, examples by oracle
- https://www.youtube.com/watch?v=ET2KOWTXaMI - objective look at GraalVM
- https://www.youtube.com/watch?v=mJcMM3wZA20&t=1893s - and even more [...pesymistic]
- https://hackernoon.com/why-the-java-community-should-embrace-graalvm-abd3ea9121b5
- https://www.slideshare.net/jyukutyo/graal-in-graalvm-a-new-jit-compiler
- https://github.com/Azbesciak/TaskScheduler - repository with my native-prepared scala app