Graal, GraalVM, Truffle:

What do they mean for polyglot developers?

•••

26-27th March 2018
59th CREST Open Workshop
UCL

Presentation: live slides

http://bit.ly/graal-polyglot-devs

About me

- Software Engineer
- Interests: code quality, testing, performance, AI/ML, NN, etc...
- Strengthening teams and helping them go faster
- Data processing and source code analysis at Prodo.AI
- Involved with various developer communities
- Attends and helps co-organise various events and conferences

Mani Sarkar
@theNeomatrix369

Thank you

- Team behind CoW
- UCL
- Sponsor(s) for the event: DAASE
- Guests and attendees
- Prodo.AI
- Anyone else not named...

Disclaimer

- Research work done by others
- Tooling around the concepts and topics
- Experimental and bleeding edge, not for prod yet
- Concise, covering surface material
- Lots of additional resources shared
- Any contributions / feedback is welcome

Agenda

- Background
  - familiarity...
  - terminologies...
  - what, how, ... in a nutshell
- Hands-on / demo
  - single languages
  - embed
  - native image
  - other fun stuff
- In production
- Summary

Familiarity

Knows about or uses Java / JVM languages?

Knows about how the JVM works?

Familiarity

Who knows about Graal, GraalVM, Truffle?

Played with it prior to this session?

Background: terminologies

What is polyglot?

What is a polyglot developer?

Background: terminologies

What is polyglot?

In computing, a polyglot is a computer program or script written in a valid form of multiple programming languages, which performs the same operations or output independent of the programming language used to compile or interpret it. [1]

https://en.wikipedia.org/wiki/Polyglot_(computing)

Background: terminologies

What is a polyglot developer?

A developer who knows and writes code in multiple languages (more a generalist) than a specialist (writes in one language only).

One who uses multiple languages regularly.

https://blog.lelonek.me/be-a-polyglot-programmer-6e7423916ed8

Background: what is Hotspot VM?

- A virtual machine that runs Java byte code
- Supports multiple platforms and operating systems
- Has an online compiler: JIT
- Has a profiler: monitors code executions
  - helps make optimisation and de-optimisation decisions
- Blackbox:
  - written in C/C++, hard to read, modify or extend the code
  - tightly coupled components, may have bugs
  - interop can be harder and expensive

Background: HotSpot JVM and JIT compiler

- Java 9 onwards
  - JVMCI: Java-level JVM Compiler Interface
    - Pluggable JIT
- Java 8
  - Custom JVM: downloadable from Oracle OTN

Background: What is Graal?

In short: a dynamic **compiler** written in Java to replace the JIT compiler in the **HotSpot VM**. A better implementation of the C2 compiler.

Broader sense: a project for developing a JIT compiler and the polyglot runtime for the HotSpot JVM i.e. Graal compiler

Background: What is GraalVM?

- Traditional HotSpot VM (optimised)
- Graal compiler replacing C2 in HotSpot
- Ability to run Truffle-enabled languages
- Additional polyglot tooling

About GraalVM

Background: What is Truffle?

Truffle is a framework for implementing languages and instruments that use Graal as a dynamic compiler.

Uses its own internal representation (IR) to build ASTs.

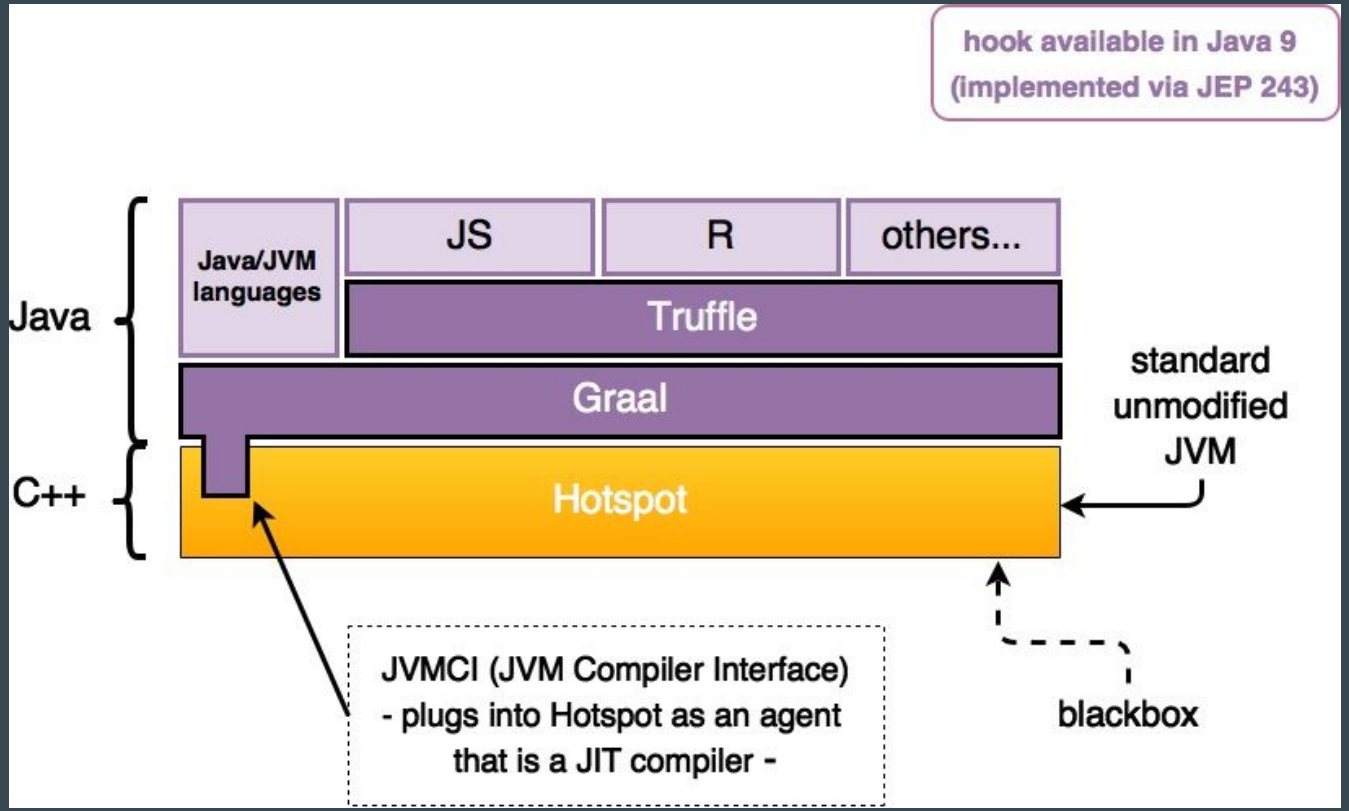
Background: languages supported on the GraalVM

SuLong (C/C++, Fortran, other languages that can be transformed to LLVM bitcode),

Java (of course),

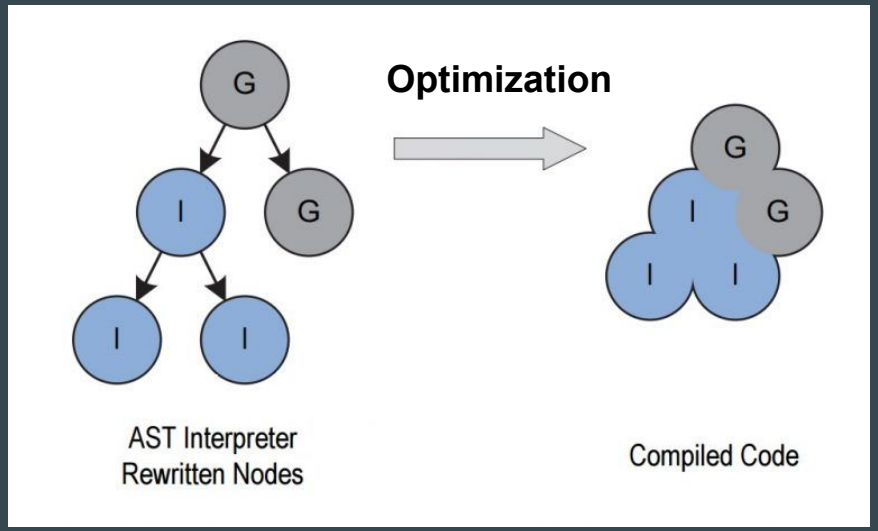
JavaScript, Ruby, Python and R

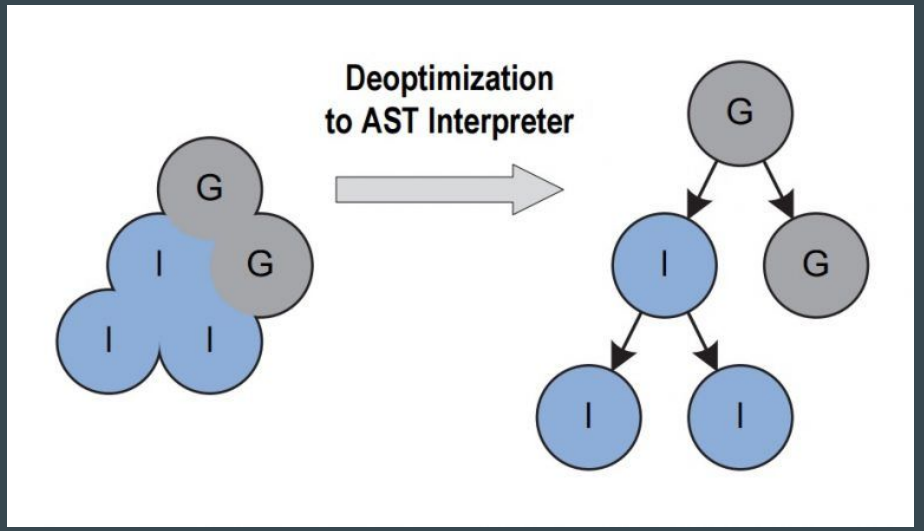
Background: In a nutshell



Graal/GraalVM: ASTs as first class citizen

Performance

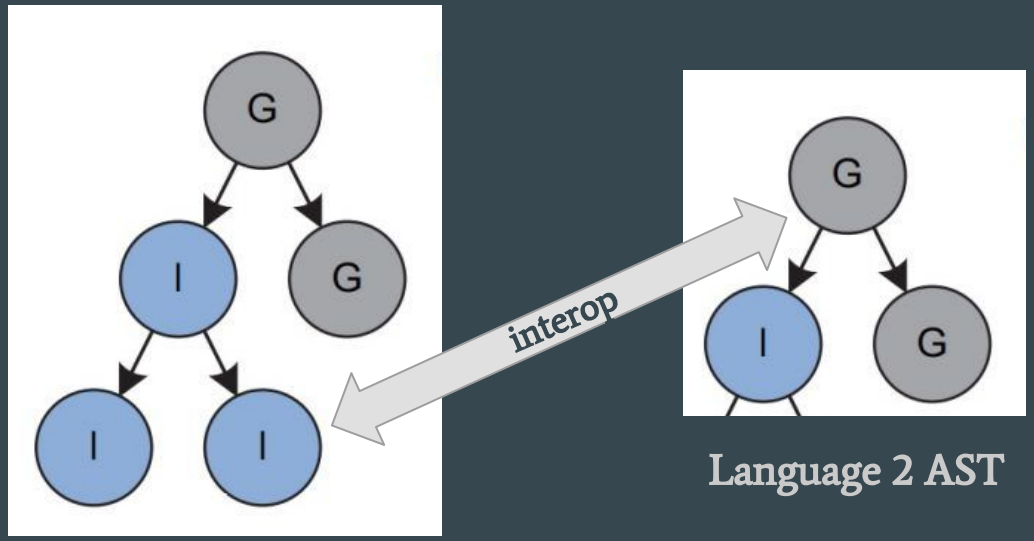




See Truffle served in a Holy Graal: Graal and Truffle for polyglot language interpretation on the JVM

Graal/GraalVM: ASTs as first class citizen

Language inter-op

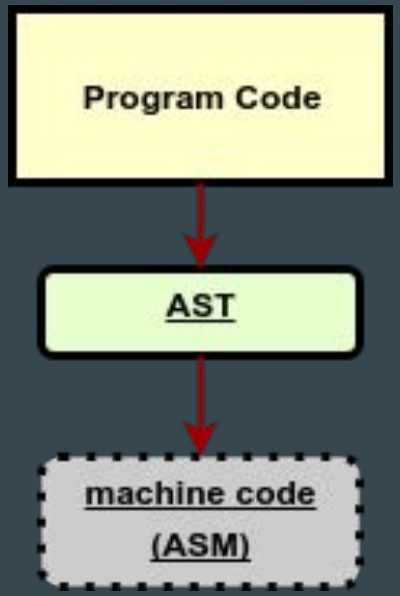


Language 1 AST

Language 2 AST

In essence there is no concept of languages at the Graal/GraalVM levels, its ASTs all the way...

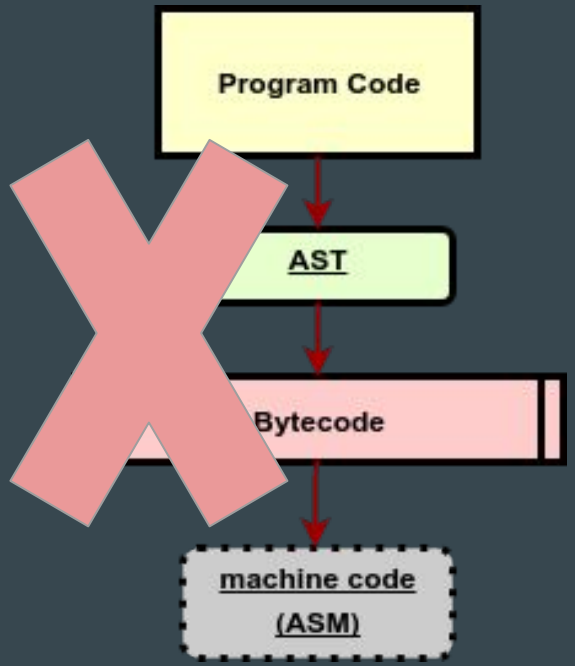
Graal/GraalVM: ASTs as first class citizen



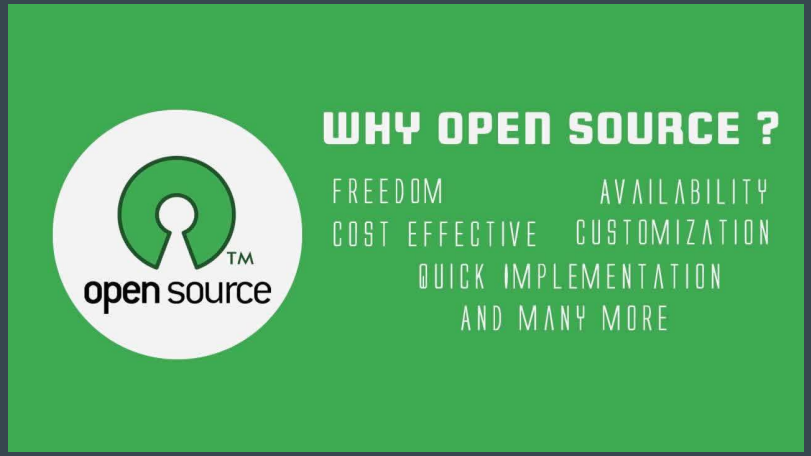
Graal/GraalVM work with ASTs,

NO bytecode translation step is necessary.

ASTs are mapped to platform/OS *specific machine code*



Graal/Truffle



Combination of GPL 2 and GPL 2 with Classpath exception

Hands on / demo

- single languages
- embed
- native image
- other fun stuff

Polyglot examples

- https://github.com/graalvm/examples/
- Weather predictor: is an application that performs temperature prediction using Ruby, R and Node.js
- https://github.com/graalvm/graal-js-archetype
  - https://github.com/oracle/graal/blob/master/sdk/docs/PolyglotEmbedding.md

In Production

@Twitter are the first brave users

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

To reach out to the VM team at Twitter:

Tweet with #TwitterVMTeam

Summary

- What do the terms mean?
- Future potential
- Hands on / demo
- Example of usage in production
- Research material
- Resources to take away

Resources

- Glossary of terms
  - to follow soon...
- Other resources
  - Blog posts: post 1 | post 2 | post 3
  - http://github.com/neomatrix369/awesome-graal.git

Citations

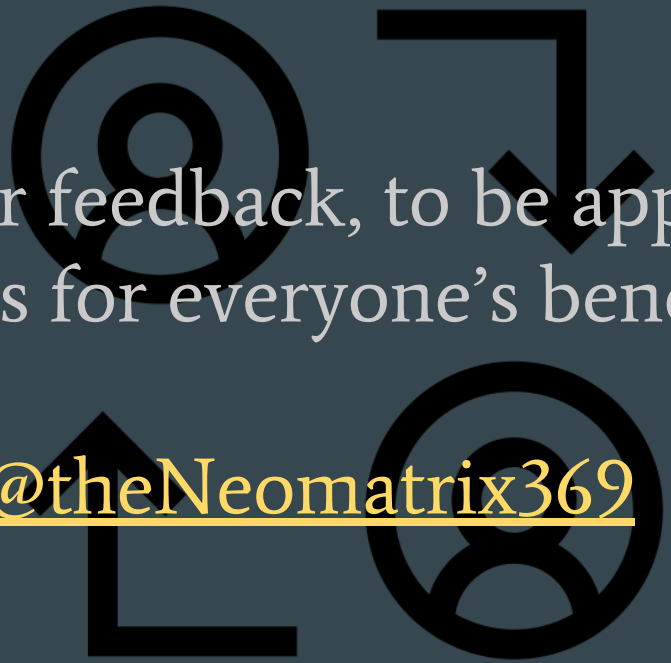
Some of images used in this presentation are owned by the respective authors, and most of them come from the https://thenounproject.com.

Also citing the diagrams in the previous slides: they have been re-used from the paper One VM to Rule Them All, the authors to credit are *Thomas Wurthinger, Christian Wimmer, Andreas Woß, Lukas Stadler, Gilles Duboscq, Christian Humer, Gregor Richards, Doug Simon and Mario Wolczko.*

Thank you, again...

- Team behind CoW
- UCL
- Sponsor(s) for the event: DAASE
- Guests and attendees
- Prodo.AI
- Anyone else not named...

Feedback and contact



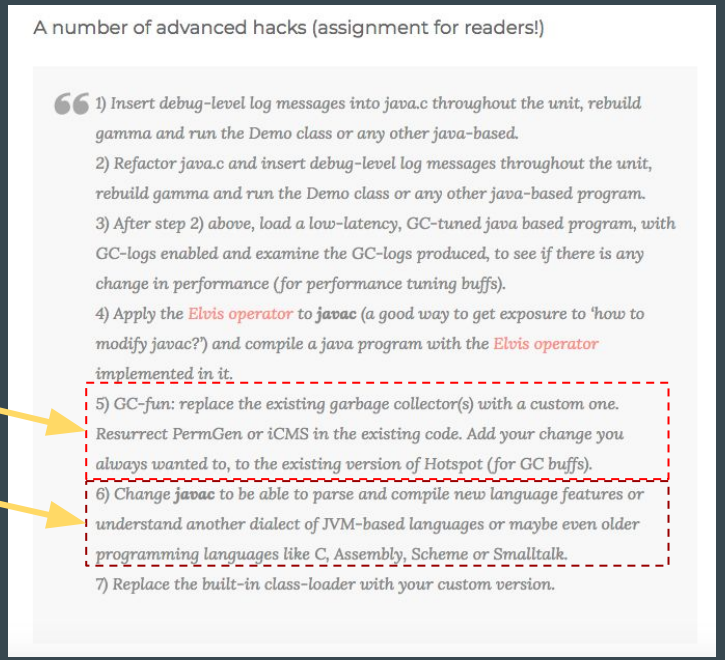
Please share your feedback, to be applied to the live slides for everyone’s benefit

@theNeomatrix369

Appendix

History: peek into the past

- Blog post written 5 years ago
  - Challenges and ideas
- At least a couple of them have be seen implemented...



JSRs / JEPs involved

JSR 223: Scripting for the Java Platform (merged as part of the Java 9 release)

JEP 243: Java-Level JVM Compiler Interface

Truffle: how to write your own language?

A simple example language built using the Truffle API
https://github.com/graalvm/simplelanguage

Interop (kind of): an extended HelloWorld example

```
import org.graalvm.polyglot.*;

public class HelloPolyglotWorldInterOp {

    public static void main(String[] args) throws Exception {
        System.out.println("Hello polyglot world Java!");
        Context context = Context.create();

        // Javascript
        Value jsReturn = context.eval("js", "function(x) (x ^ 2) + 1");
        System.out.println("Returned value (js object): " + jsReturn);
        int js = jsReturn.execute(41).asInt();
        System.out.println("Returned value (js function execution): " + js);

        // Ruby
        Value rubyReturn = context.eval("ruby", "1 + 2");
        int ruby = rubyReturn.asInt();
        System.out.println("Returned value (ruby evaluation): " + ruby);

        // R
        Value rReturn = context.eval("R", "6.5 + 7.2;");
        double r = rReturn.asDouble();
        System.out.println("Returned value (R evaluation): " + rReturn);

        // python
        Value pythonReturn = context.eval("python", "3 + 4");
        int python = pythonReturn.asInt();
        System.out.println("Returned value (python evaluation): " + pythonReturn);

        double total = js + ruby + r + python;
        System.out.println("Total values: " + total);
    }
}
```

What is SubstrateVM?

A framework that allows ahead-of-time (AOT) compilation of Java applications under the closed-world assumption into executable images or shared objects

https://github.com/oracle/graal/tree/master/substratevm

Limitations of SubstrateVM

https://github.com/oracle/graal/blob/master/substratevm/LIMITATIONS.md

Side note about Twitter and HotSpot

Twitter have rewritten many parts of the HotSpot VM (using internally)

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