

ORACLE®

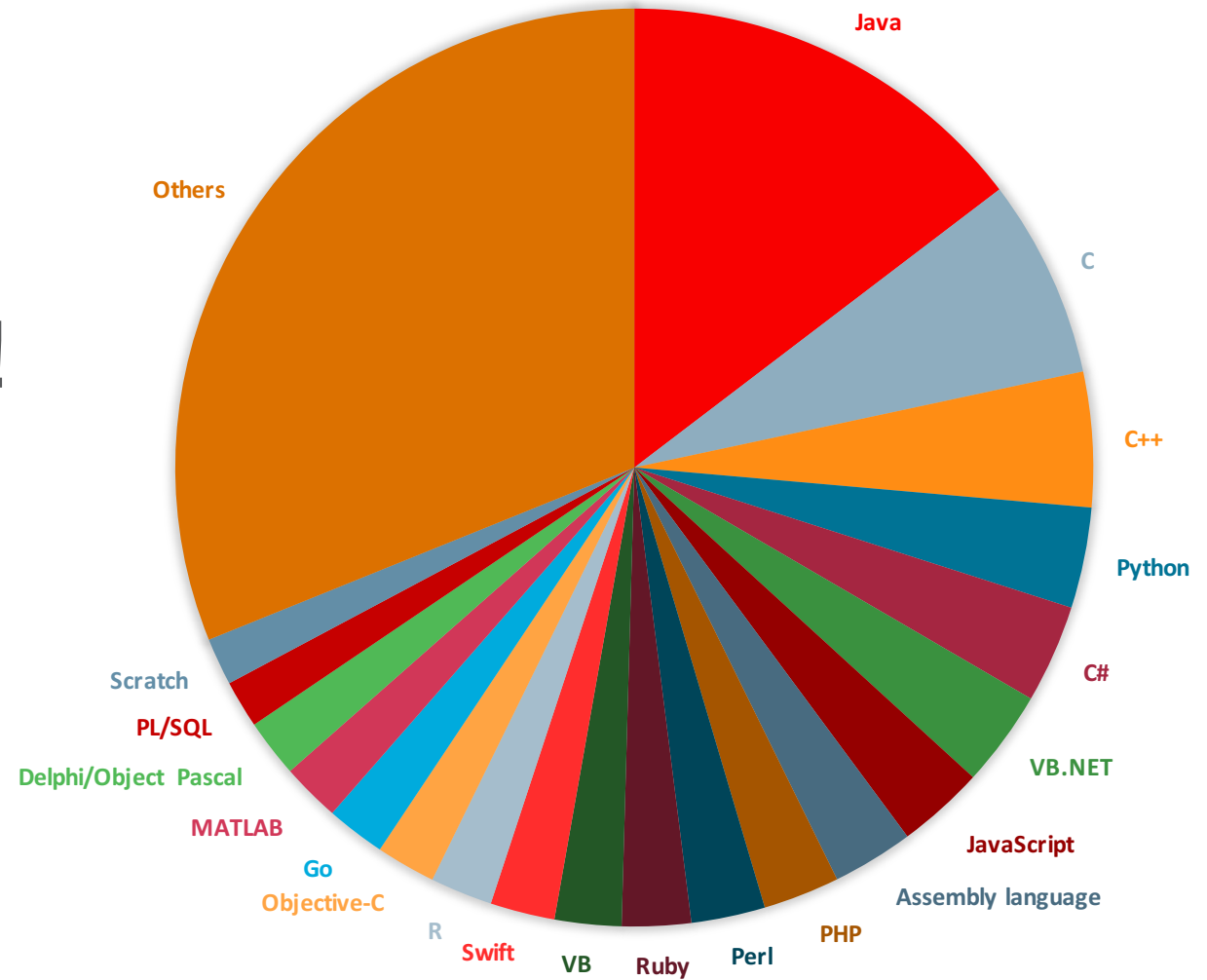
Safe Harbor Statement

The following is intended to provide some insight into a line of research in Oracle Labs. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in connection with any Oracle product or service remains at the sole discretion of Oracle. Any views expressed in this presentation are my own and do not necessarily reflect the views of Oracle.

PROGRAMMING LANGUAGE POPULARITY

(TOP 20 LANGUAGES FROM MAY 2017 TIOBE INDEX)

The World Is Polyglot!





stackoverflow

Questions

Tags

Tour

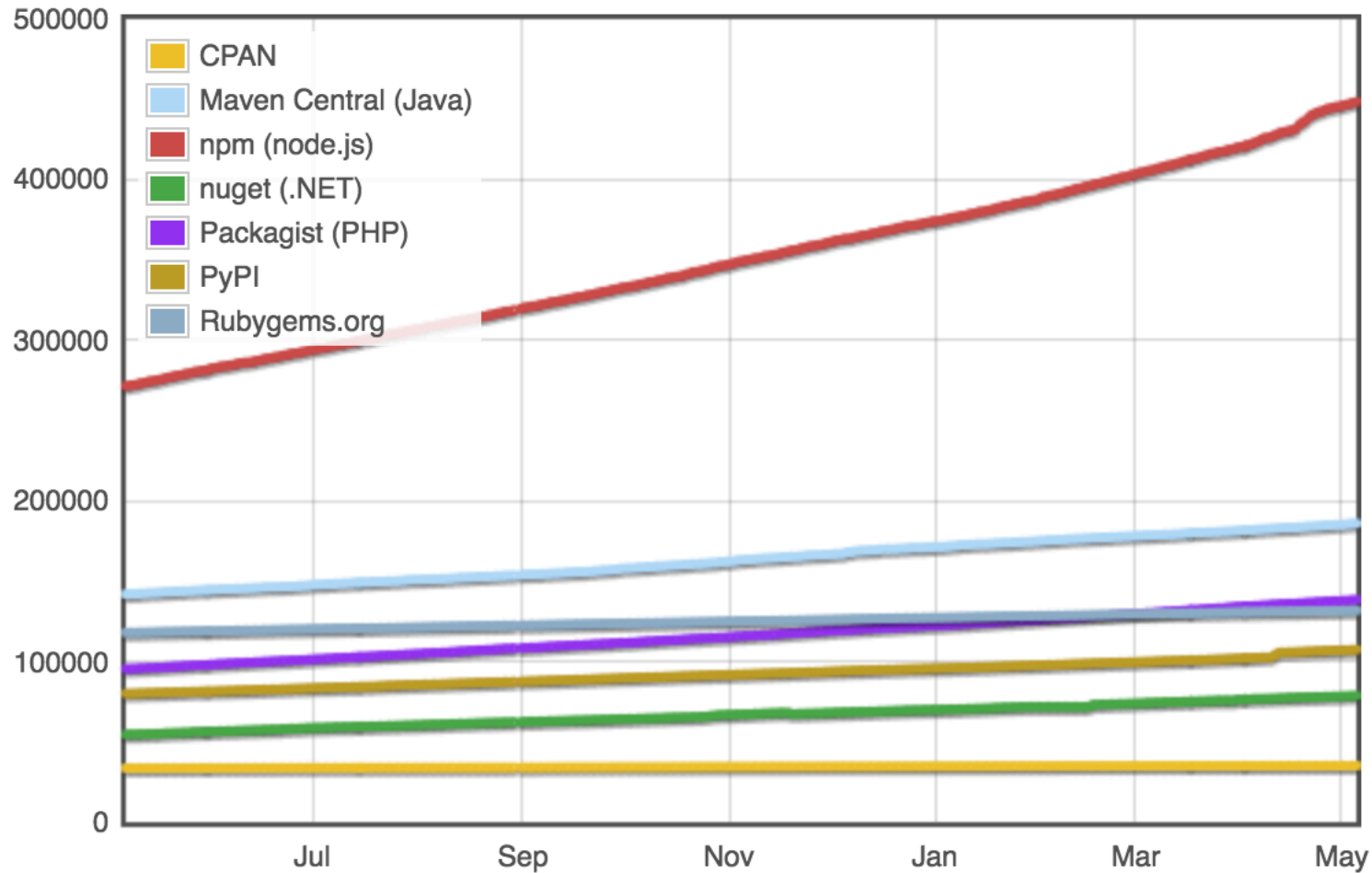
Users

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.

Why can't there be an “ultimate” programming language?

closed as not constructive by [Tim](#), [Bo Persson](#), [Devon_C_Miller](#), [Mark, Graviton](#) Jan 17 at 5:58

Module Counts



Include

- Clojars (Clojure)
- CPAN
- CPAN (search)
- CRAN (R)
- Crates.io (Rust)
- Drupal (php)
- DUB (dlang)
- Gopm (go)
- Hackage (Haskell)
- Hex.pm (Elixir/Erlang)
- Julia
- LuaRocks (Lua)
- Maven Central (Java)
- MELPA (Emacs)
- npm (node.js)
- nuget (.NET)
- Packagist (PHP)
- Pear (PHP)
- Perl 6 Ecosystem (perl 6)
- PyPI
- Rubygems.org

Polyglot Challenges

Only languages with high industry attention achieve high performance.

Languages execute in their own box with costly serialization between boxes.

Languages require different tools for configuring, debugging, profiling...

Graal Vision

High performance for all languages

Zero overhead interoperability between languages

Language-level virtualization layer for shared manageability and tooling

You can execute any language on the JVM / CLR
- as long as it looks like Java / C#.

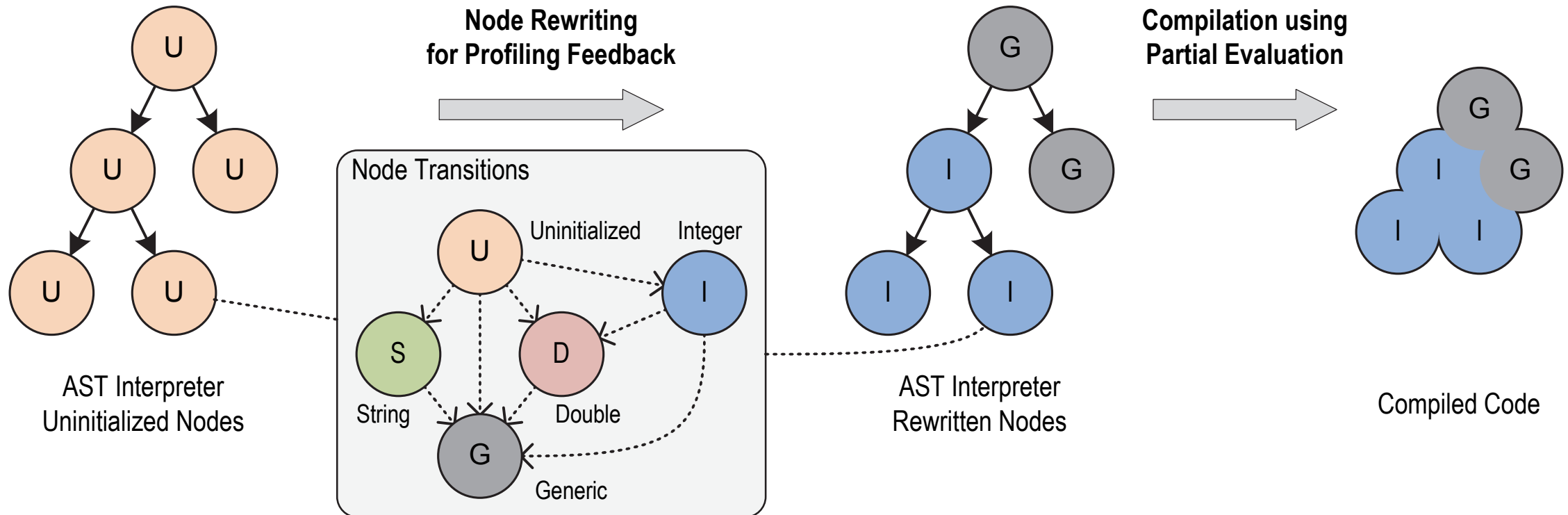
Mismatch between type system and bytecodes of host VM and semantics of guest language.

After 5+ years of research...

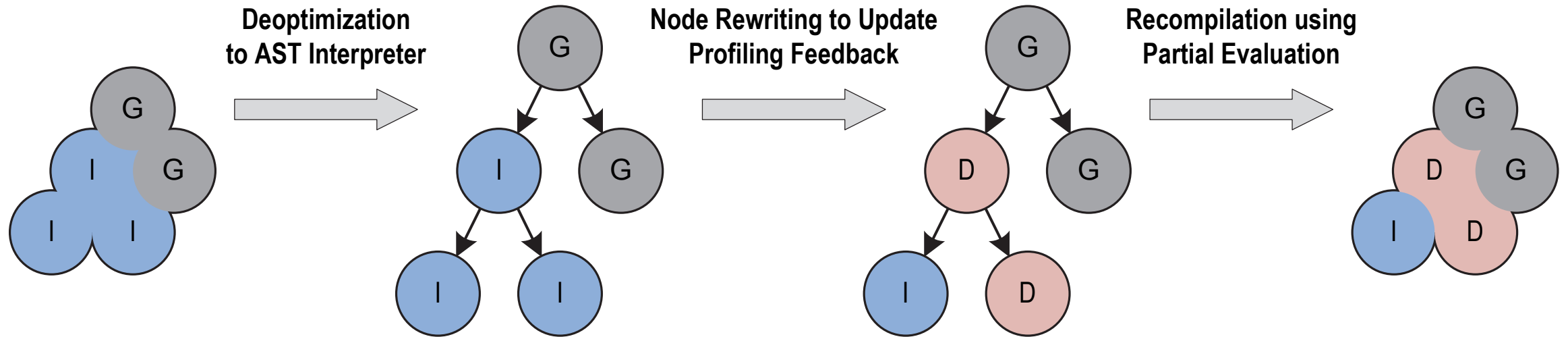
One VM to Rule Them All

Thomas Würthinger* Christian Wimmer* Andreas WöB† Lukas Stadler†
Gilles Duboscq† Christian Humer† Gregor Richards§ Doug Simon* Mario Wolczko*
*Oracle Labs †Institute for System Software, Johannes Kepler University Linz, Austria §S³ Lab, Purdue University

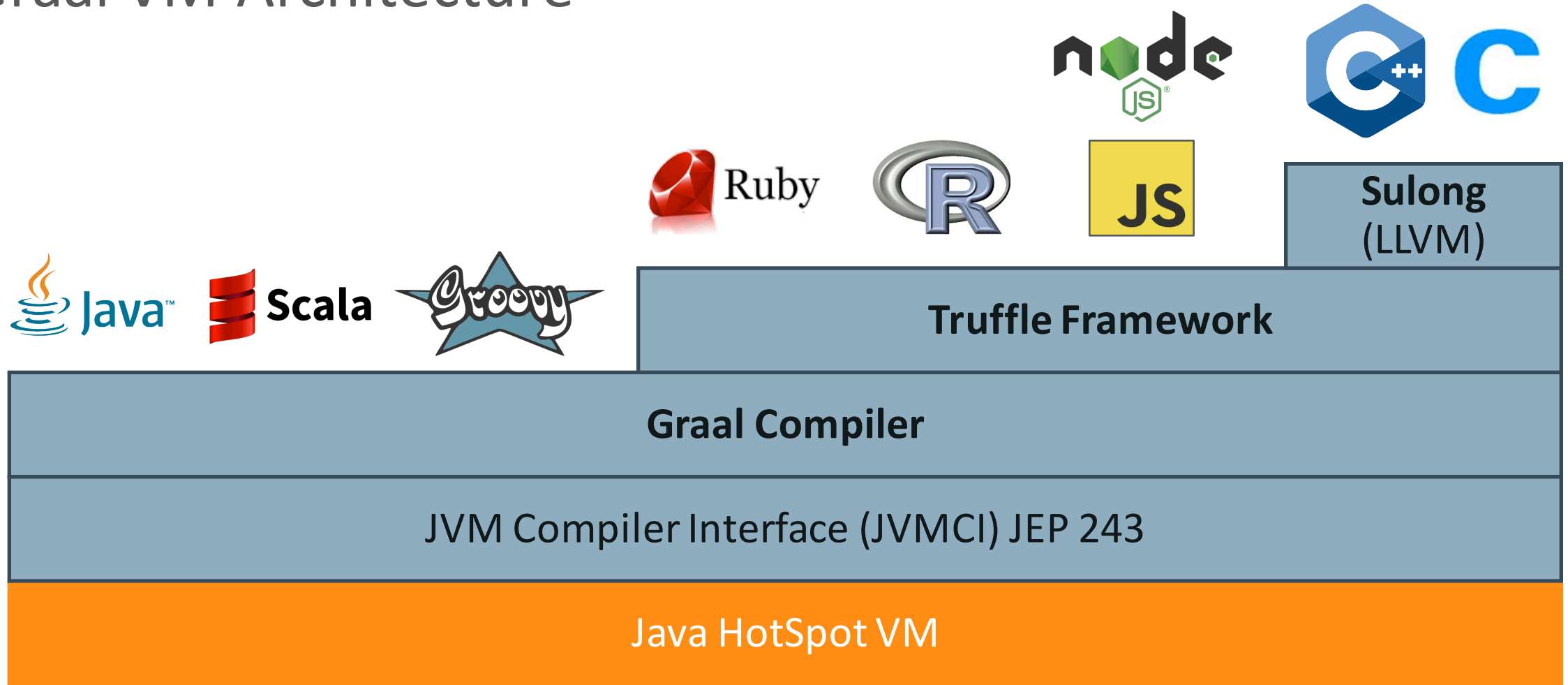
Optimization and Speculation...



And Deoptimization!



Graal VM Architecture

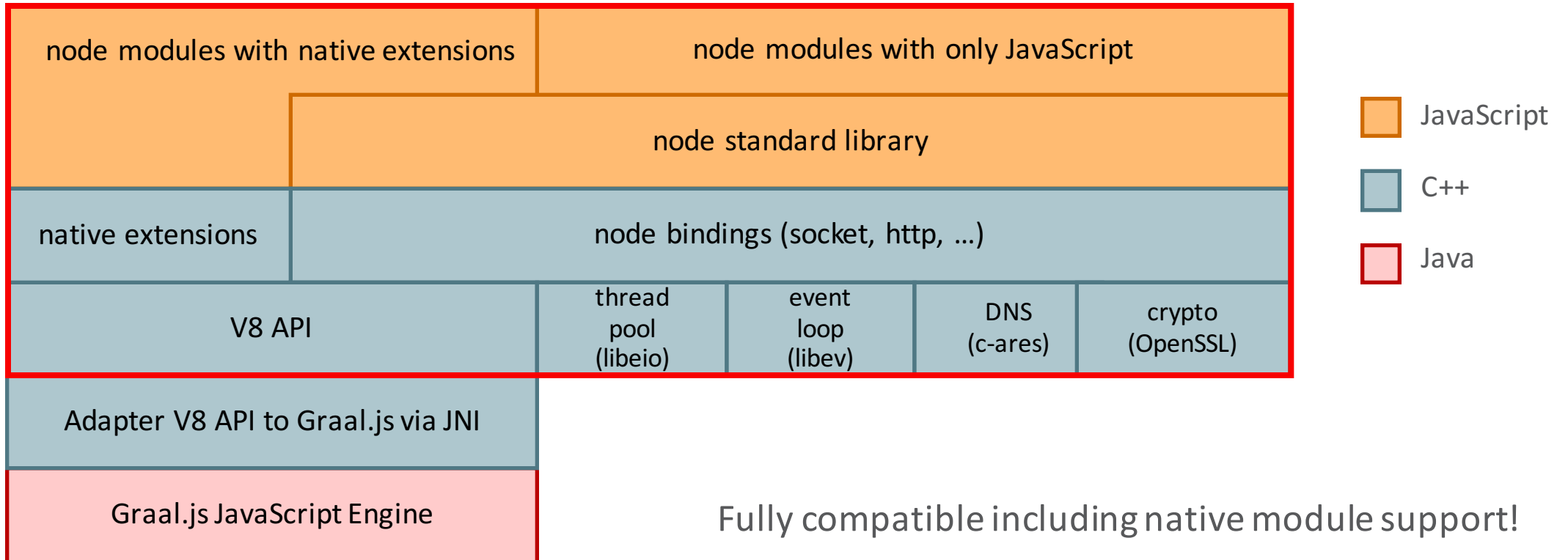


Let's talk about R...

```
function(a, x) (a[x]+42)
```

- Does „+“ resolve to the addition builtin function?
- Class of „a“?
- Does „[]“ resolve to the array access builtin function?
- Is „x“ a vector of specific length?
- Does „x“ have negative elements? The 0 element? Positive elements? In-bounds or out-of-bounds? An NA element?
- Any code can be executed at the point where „a“ or „x“ is evaluated (lazy promise evaluation of arguments)
- Does evaluated code redefine „+“ or „[]“ functions?
- Is intermediate vector result needed?

Graal.js Architecture



Fully compatible including native module support!

Project Sulong: LLVM front-end for Graal

<https://github.com/graalvm/sulong>

C/C++

```
int add(x, y) {  
    return x + y;  
}
```

Fortran

```
FUNCTION add(x, y)  
    INTEGER :: add  
    INTEGER :: a  
    INTEGER :: b  
    add = a + b  
    RETURN  
END FUNCTION
```

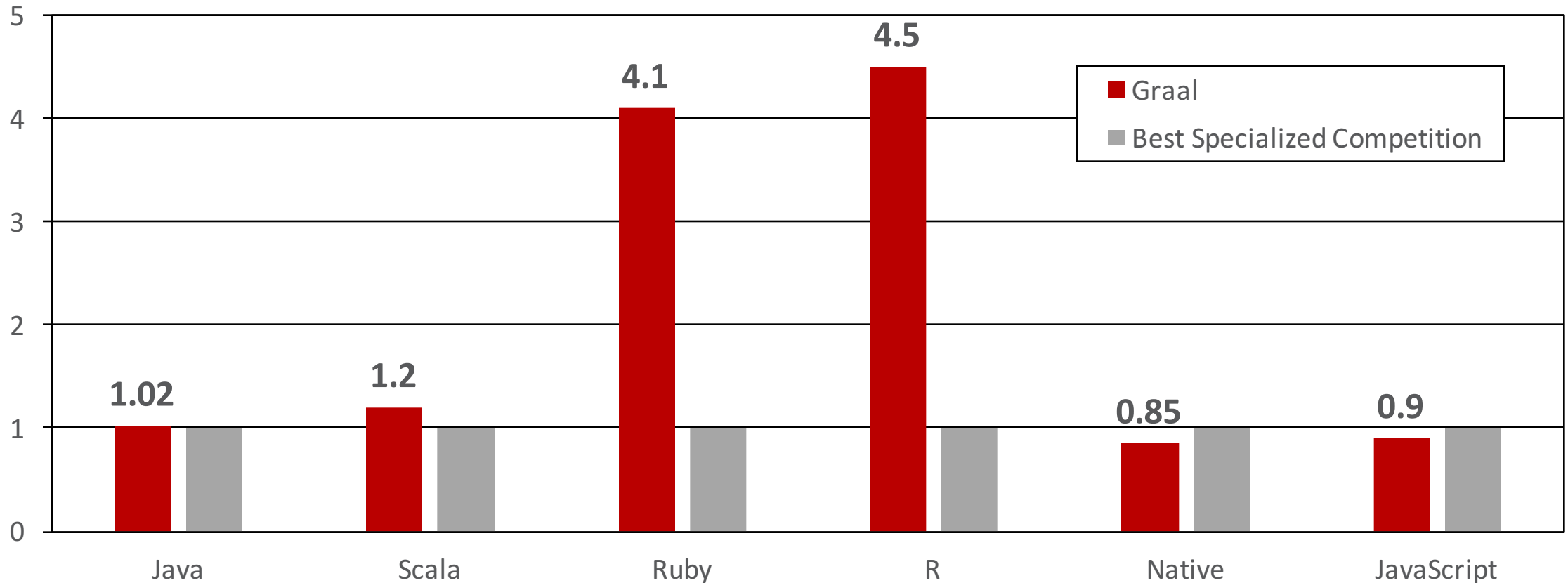
Go

```
func add(x int, y int) int {  
    return x + y;  
}
```

```
define i32 @add(i32 %x, i32 %y) #0 {  
    %1 = alloca i32, align 4  
    %2 = alloca i32, align 4  
    store i32 %x, i32* %1, align 4  
    store i32 %y, i32* %2, align 4  
    %3 = load i32* %1, align 4  
    %4 = load i32* %2, align 4  
    %5 = add nsw i32 %3, %4  
    ret i32 %5  
}
```


High-Performance for **Each** Language

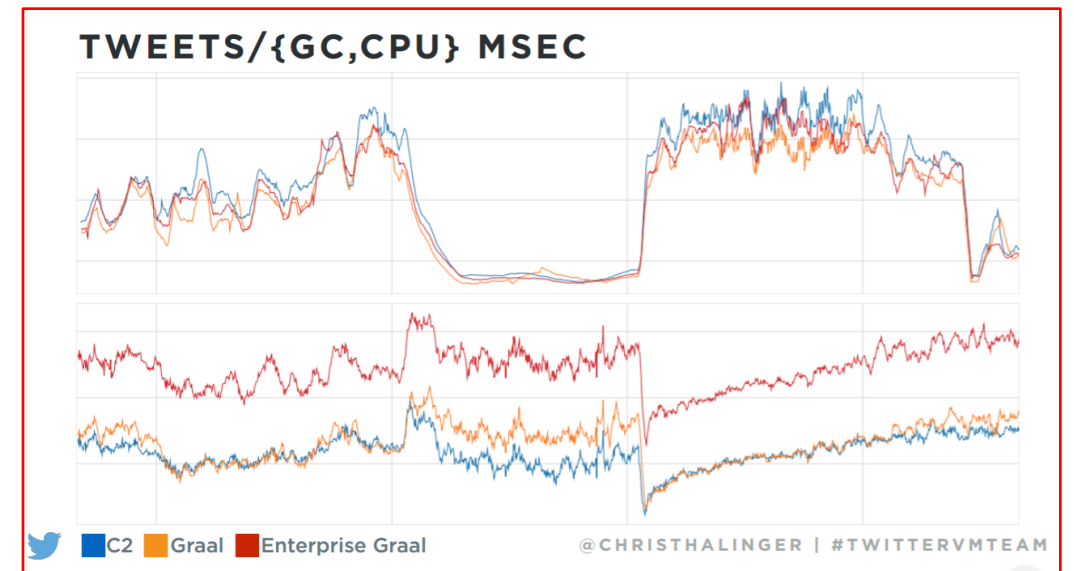
Speedup, higher is better



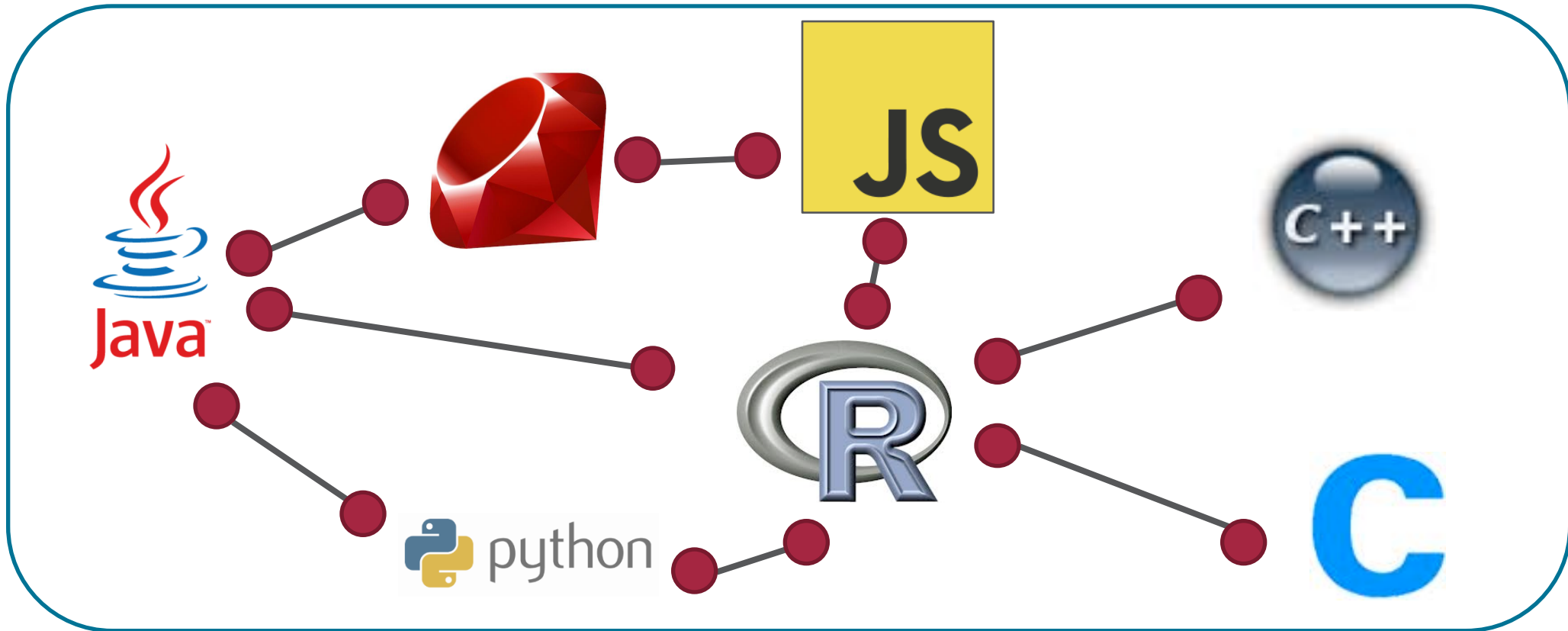
Performance on 64-bit x86 on well-known benchmark suites relative to:
HotSpot/Server, JRuby, GNU R, LLVM AOT compiled, V8

Twitter Evaluated Early Version of Graal

- Large number of Scala microservices
- More Tweets per CPU msec in their core service
- <https://www.jfokus.se/jfokus17/preso/Graal-at-Twitter.pdf>



Zero Overhead Interoperability



Graal runtime supports **complete separation of physical and logical layout** for guest language data structures.

- Defining lambda method in JavaScript:

```
function area(rect) {  
    return rect.width * rect.height;  
}
```

- Calling from same language:

```
area({width:100, height:50});
```

- Or from other language (e.g., Java):

```
class Rect { int width, height; ... }  
callJS("area", new Rect(100, 50));
```

- Or from custom memory layouts:

```
SELECT width, height from data;
```

- row-oriented format => raw pointer to current row and schema
- column-oriented format => two raw pointers to current positions within columns and schema

Data Sharing Across Language Boundaries

main.c

```
#include<stdio.h>

struct complex {
    double r;
    double i;
}

int main() {
    struct complex *a = ...;
    struct complex *b = ...;

    add(a, b)
}
```

complex.js

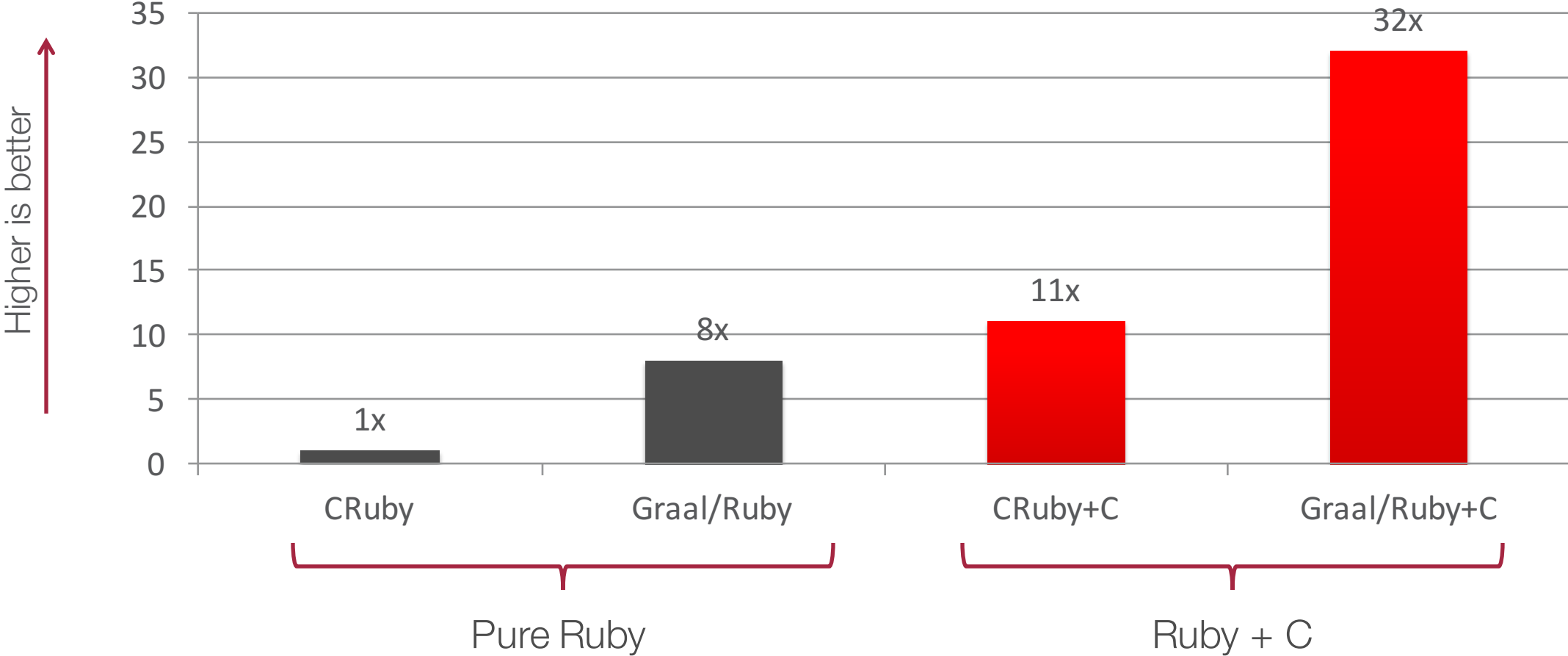
```
function add(a, b) {
    var result = {r:0, i:0};

    result.r = a->r + b->r
    result.i = a->i + b->i

    return result;
}
```



Image Processing Composite Speedup



DEMO

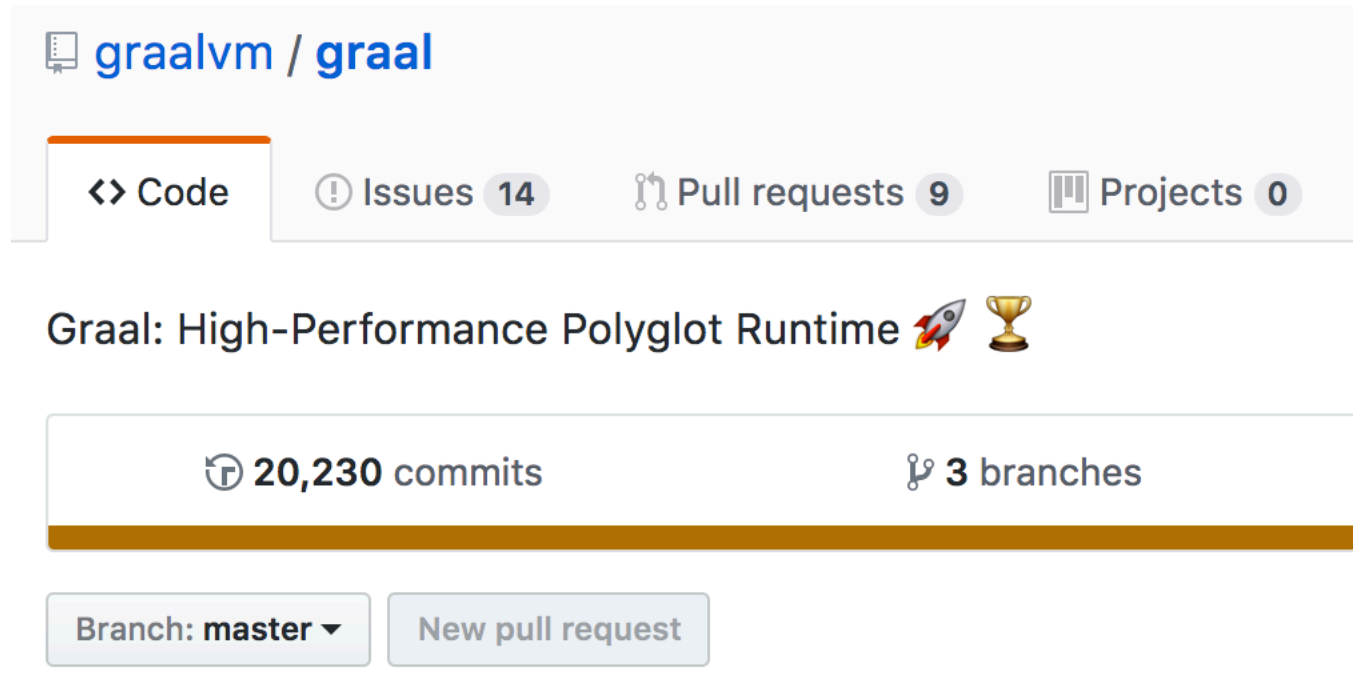
Graal Extended Vision

High performance, polyglot, language-level virtualization layer...

embeddable across the stack in native and JVM-based applications.

Resources on Graal

- OTN product page including download:
 - www.oracle.com/technetwork/oracle-labs/program-languages
- Graal projects on github:
 - github.com/graalvm
- Graal on OpenJDK:
 - openjdk.java.net/projects/graal



graalvm / graal

<> Code ! Issues 14 🔗 Pull requests 9 📁 Projects 0

Graal: High-Performance Polyglot Runtime 🚀 🏆

🔄 20,230 commits 🔗 3 branches

Branch: master ▾ New pull request

@graal

@thomaswue

Q/A