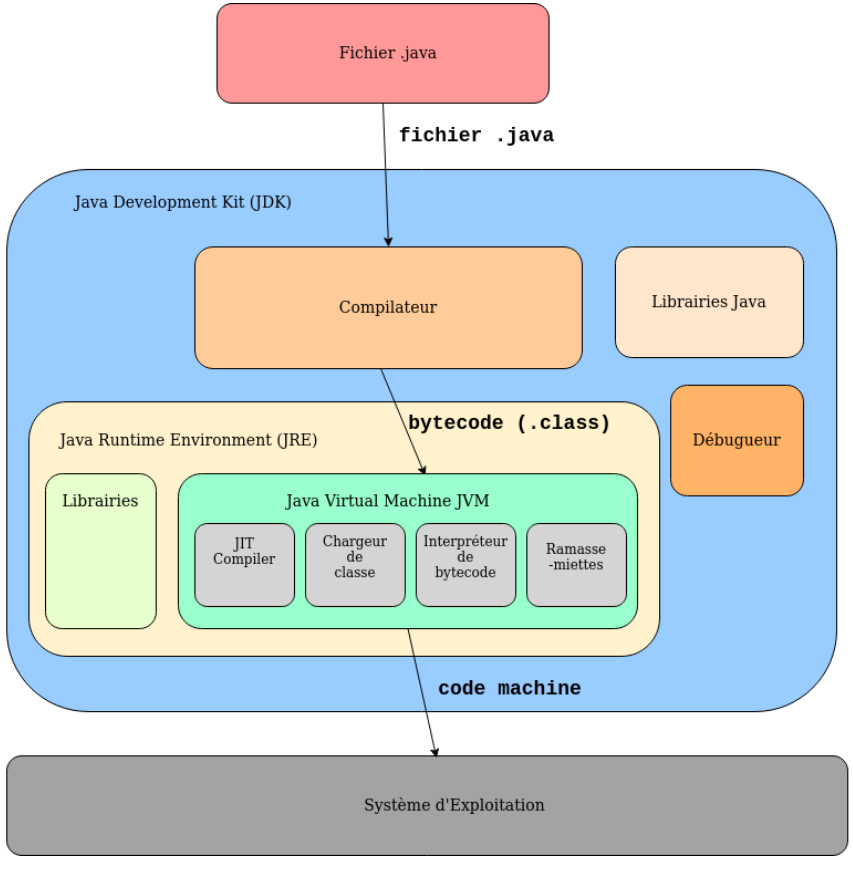
GraalVM™

LA MACHINE VIRTUELLE POLYGLOTTE D'ORACLE

RAPPELS SUR LA PLATEFORME JAVA



ORIGINES

CONSTAT

Total					
	Energy		Time		Mb
(c) C	1.00	(c) C	1.00	(c) Pascal	1.00
(c) Rust	1.03	(c) Rust	1.04	(c) Go	1.05
(c) C++	1.34	(c) C++	1.56	(c) C	1.17
(c) Ada	1.70	(c) Ada	1.85	(c) Fortran	1.24
(v) Java	1.98	(v) Java	1.89	(c) C++	1.34
(c) Pascal	2.14	(c) Chapel	2.14	(c) Ada	1.47
(c) Chapel	2.18	(c) Go	2.83	(c) Rust	1.54
(v) Lisp	2.27	(c) Pascal	3.02	(v) Lisp	1.92
(c) Ocaml	2.40	(c) Ocaml	3.09	(c) Haskell	2.45
(c) Fortran	2.52	(v) C#	3.14	(i) PHP	2.57
(c) Swift	2.79	(v) Lisp	3.40	(c) Swift	2.71
(c) Haskell	3.10	(c) Haskell	3.55	(i) Python	2.80
(v) C#	3.14	(c) Swift	4.20	(c) Ocaml	2.82
(c) Go	3.23	(c) Fortran	4.20	(v) C#	2.85
(i) Dart	3.83	(v) F#	6.30	(i) Hack	3.34
(v) F#	4.13	(i) JavaScript	6.52	(v) Racket	3.52
(i) JavaScript	4.45	(i) Dart	6.67	(i) Ruby	3.97
(v) Racket	7.91	(v) Racket	11.27	(c) Chapel	4.00
(i) TypeScript	21.50	(i) Hack	26.99	(v) F#	4.25
(i) Hack	24.02	(i) PHP	27.64	(i) JavaScript	4.59
(i) PHP	29.30	(v) Erlang	36.71	(i) TypeScript	4.69
(v) Erlang	42.23	(i) Jruby	43.44	(v) Java	6.01
(i) Lua	45.98	(i) TypeScript	46.20	(i) Perl	6.62
(i) Jruby	46.54	(i) Ruby	59.34	(i) Lua	6.72
(i) Ruby	69.91	(i) Perl	65.79	(v) Erlang	7.20
(i) Python	75.88	(i) Python	71.90	(i) Dart	8.64
(i) Perl	79.58	(i) Lua	82.91	(i) Jruby	19.84

EXISTANT

OBJECTIFS

- Accélération
- Runtime polyglotte

LA SOLUTION : GRAALVM

NOUVEAUTÉS REMARQUABLES

- GraalVM compiler
- GraalVM Native Image
- Truffle Framework et GraalVM SDK
- LLVM Runtime et JavaScript Runtime

Language Implementations

This page is intended to keep track of the growing number of Truffle language implementations and experiments. The following language implementations exist already:

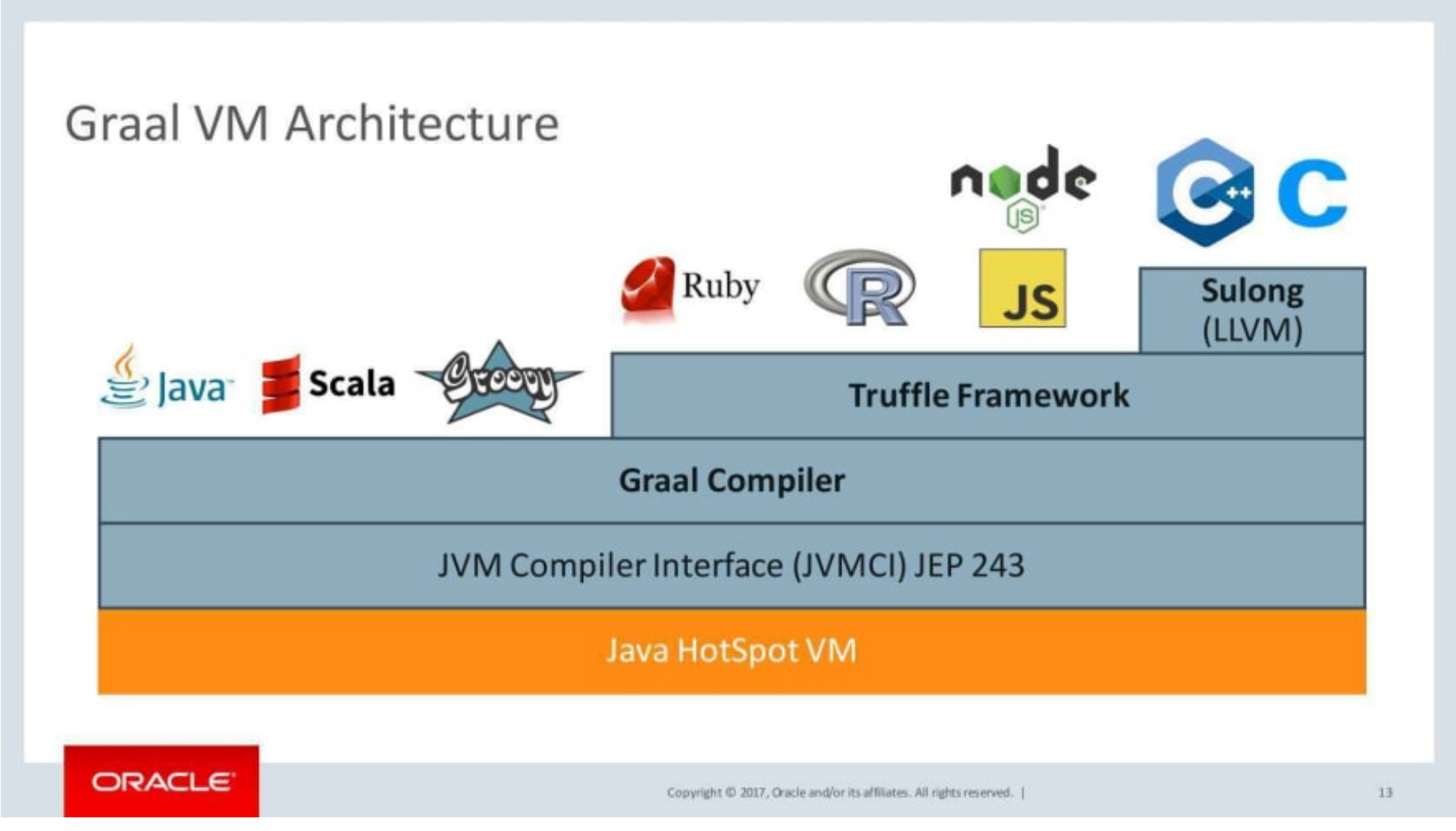
- FastR, an implementation of GNU R. *
- Graal.js, an ECMAScript 2020 compliant JavaScript implementation. *
- Graal.Python, an early-stage implementation of Python. *
- SimpleLanguage, a toy language implementation to demonstrate Truffle features.
- SOMns, a Newspeak implementation for Concurrency Research.
- Sulong, an LLVM bitcode interpreter. *
- TRegex, a generic regular expression engine (internal, for use by other languages only). *
- TruffleRuby, an implementation of Ruby. *
- TruffleSOM, a SOM Smalltalk implementation.
- TruffleSqueak, a Squeak/Smalltalk VM implementation and polyglot programming environment.
- Yona, the reference implementation of a minimalistic, strongly dynamically typed, parallel and non-blocking, polyglot, strict, functional programming language.

* Shipped as part of GraalVM.

Experiments

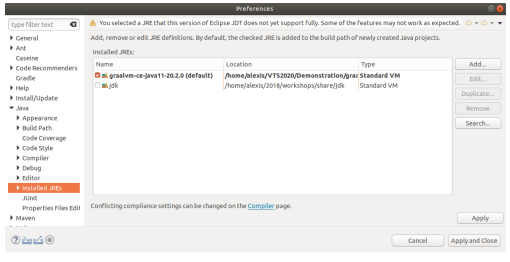
- bf, an experimental Brainfuck programming language implementation.
- brainfuck-jvm, another Brainfuck language implementation.
- Cover, a Safe Subset of C++.
- DynSem, a DSL for declarative specification of dynamic semantics of languages.
- Heap Language tutorial showing embedding of Truffle languages via interoperability.
- hextruffe, an implementation of Hex.
- LuaTruffle, an implementation of the Lua language.
- Mozart-Graal, an implementation of the Oz programming language.
- Mumbler, an experimental Lisp programming language.
- PorcE, an Orc language implementation.
- ProloGraal a Prolog language implementation supporting interoperability.
- PureScript, a small strongly typed programming language.
- Reactive Ruby, TruffleRuby meets Reactive Programming.
- shen-truffle, a port of the Shen programming language.
- TruffleMATE, a Smalltalk with a completely reified runtime system.
- TrufflePascal, a Pascal interpreter.
- ZipPy, a Python implementation.

ARCHITECTURE



UTILISATION

- Télécharger GraalVM sur le git officiel
- Remplacer le path du JDK par le path de GraalVM dans l'IDE :



OU

Launchers via ligne de commande :

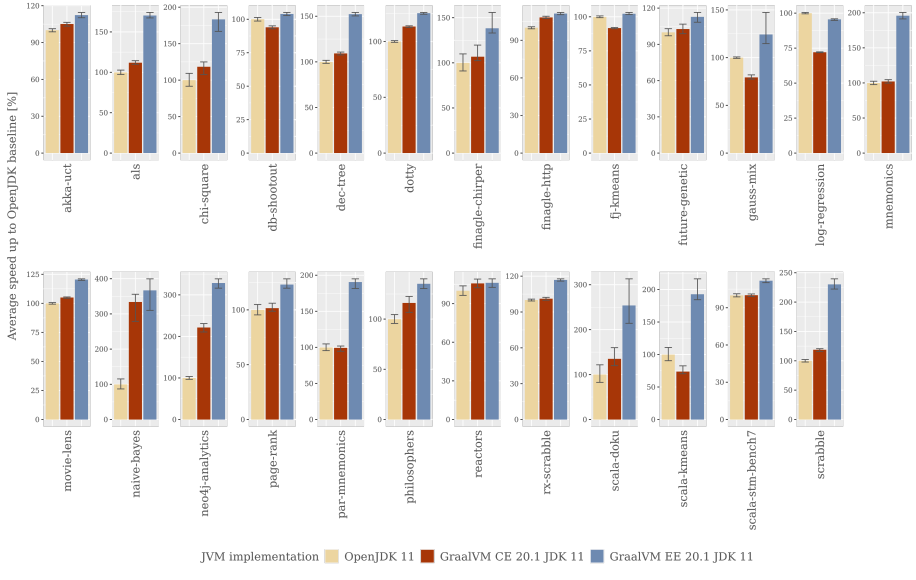
```
javac prog.java
python prog.py
```

- Librairies spécifiques aux JDK Graal

```
import org.graalvm.polyglot.*;  static Context context = Context.newBuilder().allowAllAccess(true).build();
```

```
Value function  = context.getBindings("python").getMember("my_function_in_python");
```

RÉSULTATS ET LIMITES



- Performances polyglottes mitigées
- Limitation des librairies

DÉMONSTRATION

MERCI DE VOTRE ATTENTION !

SOURCES

https://www.graalvm.org/docs/getting-started/
https://medium.com/@nelvadas/a-la-d%C3%A9couverte-de-graalvm-5adbe0487624
https://fr.wikipedia.org/wiki/Machine_virtuelle_Java
https://fr.wikipedia.org/wiki/Environnement_d%27ex%C
https://www.theserverside.com/definition/GraalVM
https://jaxenter.com/energy-efficient-programming-languages-137264.html
https://dev.to/abdulfousan/graalvm-polyglot-virtual-machine-4mn2
https://www.graalvm.org/docs/introduction/
https://renaissance.dev/