

CS614: Advanced Compilers

JIT Compilation (Cont.)

Manas Thakur
CSE, IIT Bombay



Spring 2025

Contextual Compilation



Manas Thakur

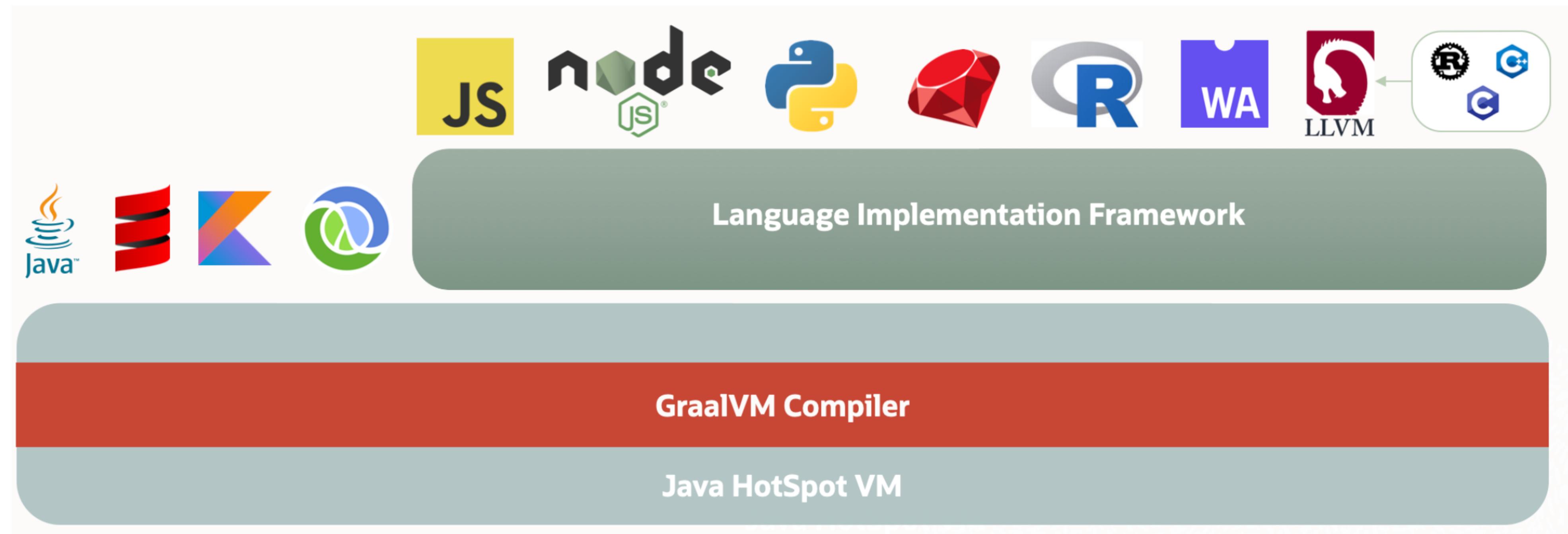
The Graal Compiler

- A separate Oracle project that created a new Java JIT compiler written in Java itself
- Can be used on top of the HotSpot JVM
- Brings in modern algorithms from research community, for Java, in Java
 - Partial escape analysis
 - Polymorphic inline caching
 - Integration with ahead-of-time (AOT) compilation
- Replace C2 with Graal and add language interoperability: **GraalVM™**



One VM to Run Them All

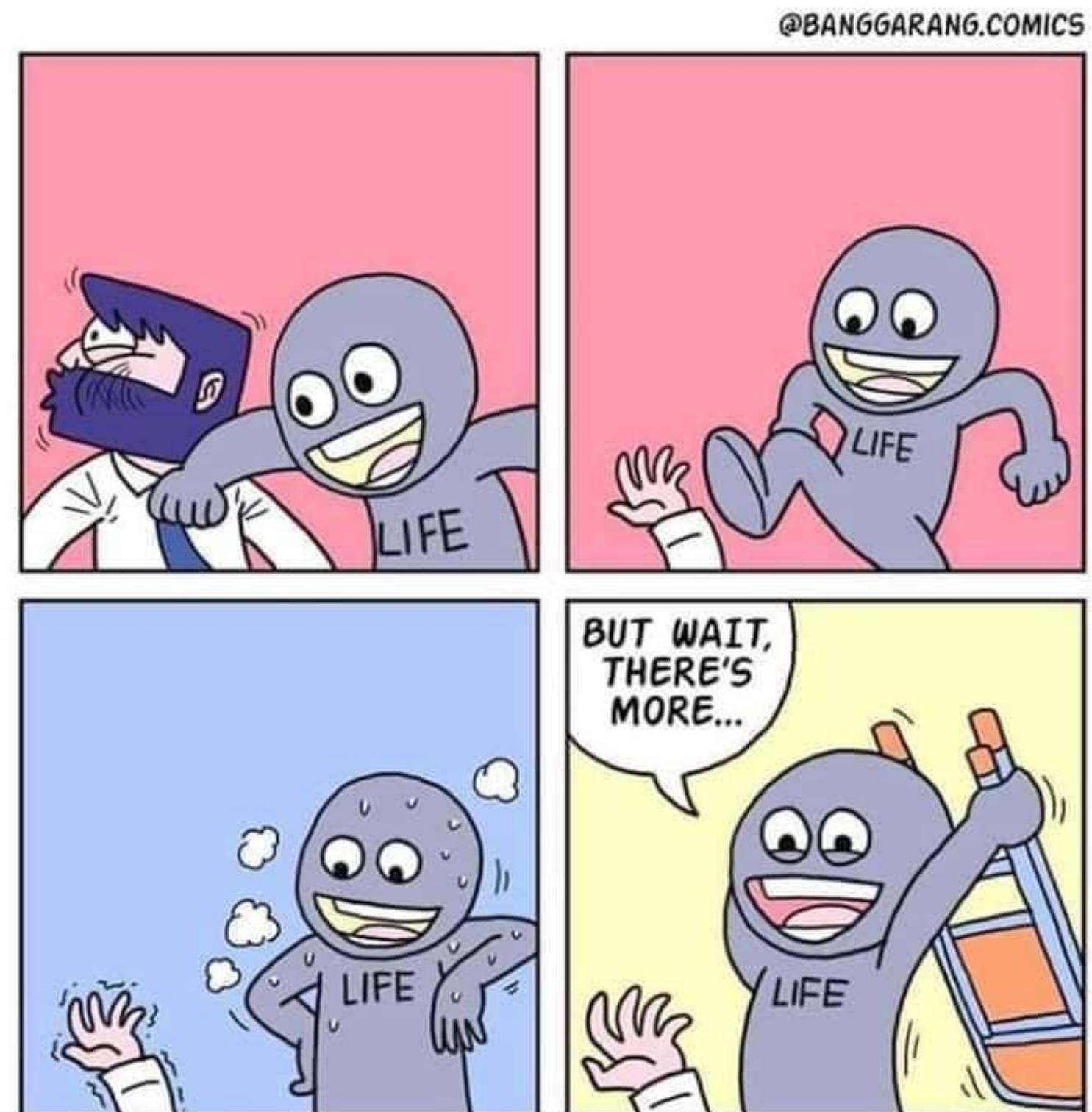
- GraalVM, an Oracle+Academia initiative, provides the Graal JIT compiler for performance and the Truffle language implementation framework for non-Java languages.



- A popular and useful approach for “polyglot programming”.

All interesting. All good. But wait!

- Time spent in JIT compilation directly affects the execution time of the program.
- Most JITs perform very imprecise analyses and lose opportunities for optimization.
- Will `gcc -Oxyz` beat standard JVM invocation?
- How to bring decades of advancements in static program analysis to the JIT world?



Active JIT Research Directions

- Newer analysis algorithms
- Better ways to enhance precision
- Reducing JIT overhead (warmup; cold start)
- Reducing deoptimization slowdowns
- Reducing profiling and deoptimization overheads
- Performing newer aggressive speculative optimizations
- Handling more dynamic features, and handling them without losing ~~much~~ precision
- Frameworks to optimize modern dynamic languages



Interested? Join CompL@PLATO

REFLECTIONS



- Theory + Practice
- The “real” world
- Punctuality
- Sincerity
- Honesty
- Rigor
- Discipline
- Novelty
- Empathy
- Technical excellence

Question of the Semester

- Which programs are the fastest?
 - Python
 - C
 - C++
 - Java

Thank
You!



It's the **compiler** that
makes programs fast.

Who writes compilers?

It's **you** who will
make programs fast.