Lambda, the Dragon Slayer!

Paulo Matos
pmatos@igalia.com
Igalia, S.L.

March 18,2023

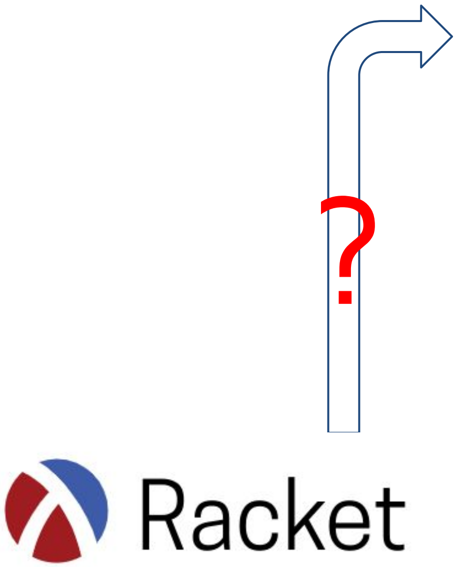
What is today about?

1. What’s this new implementation about!
2. Why MLIR/LLVM and what are these things?
3. Architecture and Plan
4. What’s the current status?
5. What’s next?
6. What to expect?
7. Where to go for more information!

Meet NORA!

- Alternative implementation of Racket
- Focused on the professional developer
- Attempt to create a community-led implementation
- but…

LLVM



```
define dso_local i32 @square(i32 noundef %0) {
 %2 = mul nsw i32 %0, %0
 ret i32 %2
}
```

LLVM IR

```
square:
# @square
       mov     eax, edi
       imul    eax, edi
       ret
```

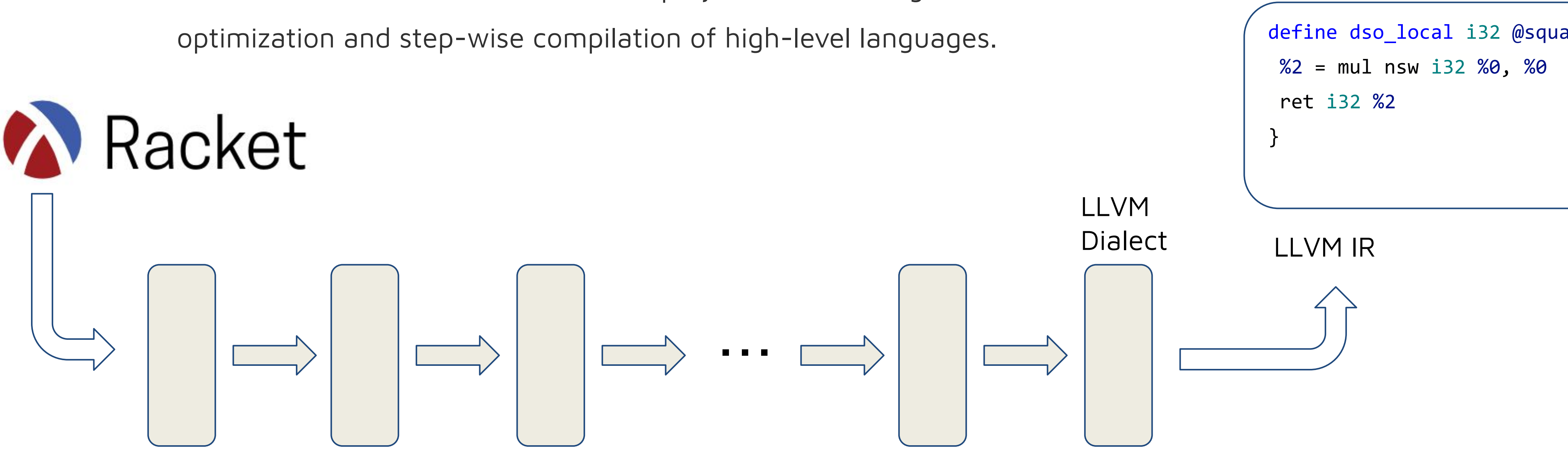
Assembly

LLVM provides:

- an IR
- a set of optimization passes
- assembler, linker, libc… etc
- API

MLIR

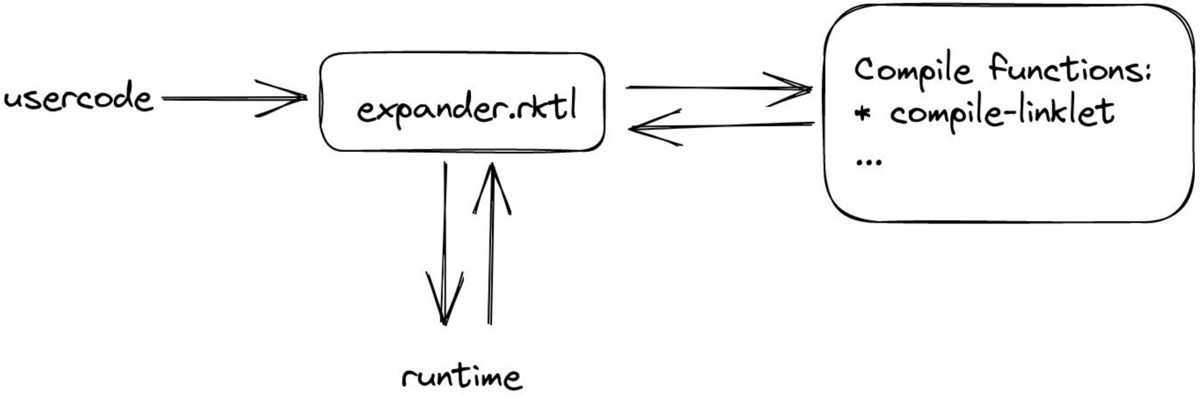
- MLIR is a recent addition to the LLVM project - one of its goals is to facilitate the optimization and step-wise compilation of high-level languages.



```
define dso_local i32 @squa
 %2 = mul nsw i32 %0, %0
 ret i32 %2
}
```

Architecture/Plan

- One of the complexities of Racket is the expander sub-system.
- Expansion is *not* a compilation phase, the expander *is* the compilation driver!
  - `expander.rktl` (the expander linklet) provides us with functions we can *call* to expand and compile user code.



But expander.rktl is a Racket linklet! How do we bootstrap it?
via an Interpreter!

What’s completed?

- Lexer is completed;
- Parser is done together with interpretation;
  - i.e. as soon as you parse something, the interpretation procedure and tests are added.
- Initial MLIR dialect NIR files are there;
  - but unused as of today.
- What can we parse/interpret?
    - arithmetic addition, define-values, identifiers, integers, lambdas, linklets and values.

What's next?

- Complete the parser and interpreter;
  - then the real work begins!
- Start thinking about the dialects necessary to lower Racket to the LLVM dialect;
  - There are other functional language examples we can take inspiration from like RISE and Lean4.

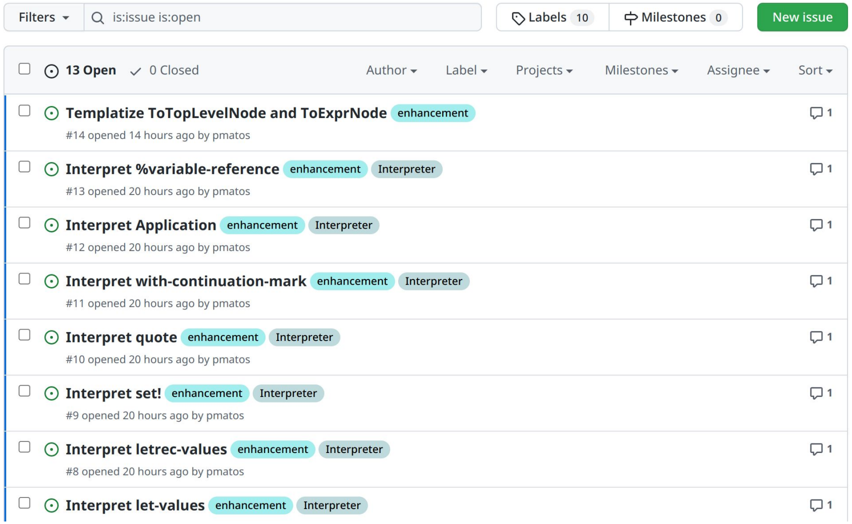
What to expect?

- If you don’t help, don’t expect much!
- Things will move as time permits.

But the goal is a compatible Racket implementation!

For more info or contribute

- □ Check out https://github.com/pmatos/nora
- □ Join the GitHub discussions
- □ Feel free to DM me on Racket discourse
- □ or drop me an email pmatos@igalia.com



□ Templatize ToTopLevelNode and ToExprNode enhancement 1
#14 opened 14 hours ago by pmatos

□ Interpret %variable-reference enhancement Interpreter 1
#13 opened 20 hours ago by pmatos

□ Interpret Application enhancement Interpreter 1
#12 opened 20 hours ago by pmatos

□ Interpret with-continuation-mark enhancement Interpreter 1
#11 opened 20 hours ago by pmatos

□ Interpret quote enhancement Interpreter 1
#10 opened 20 hours ago by pmatos

□ Interpret set! enhancement Interpreter 1
#9 opened 20 hours ago by pmatos

□ Interpret letrec-values enhancement Interpreter 1
#8 opened 20 hours ago by pmatos

□ Interpret let-values enhancement Interpreter 1 igalia