ClangIR's Footprint

A quick compile-time impact report

Bruno Cardoso Lopes

Meta

High-level IR for Clang

- Closer to C/C++ than LLVM
- Translated out of Clang’s AST
- Move Clang onto the MLIR substrate
- LLVM incubator: currently 27% (lines of text) upstreamed (Jun 25’)

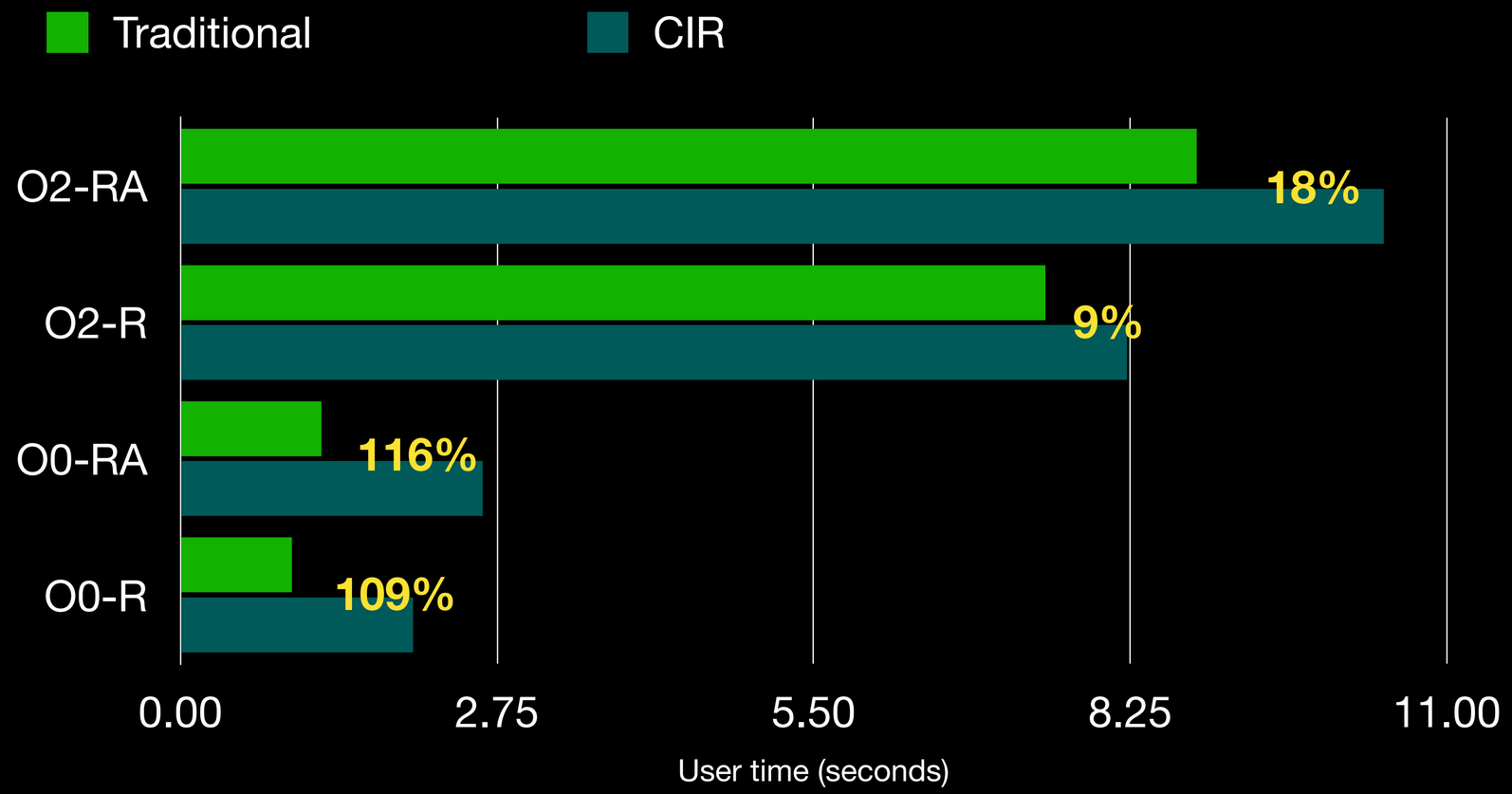
Benchmarks

Current status llvm-testsuite

- SingleSource: 1665 / 1833 (91%) pass
- MultiSource: 161 / 201 (80%) pass
- CTMark: 3/10 (30%) pass

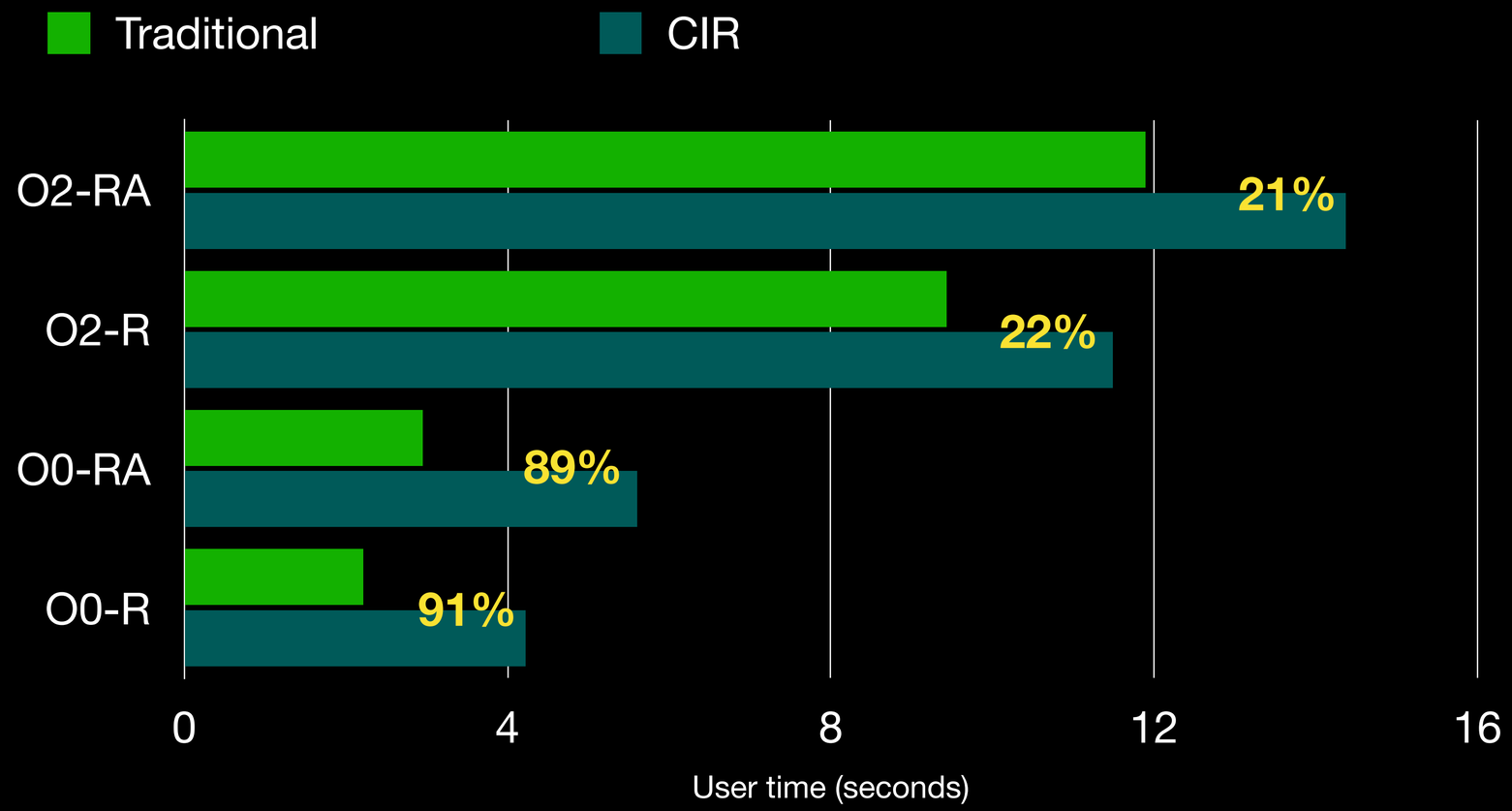
CTMark

mafft/pairlocalalign (27 C source files)



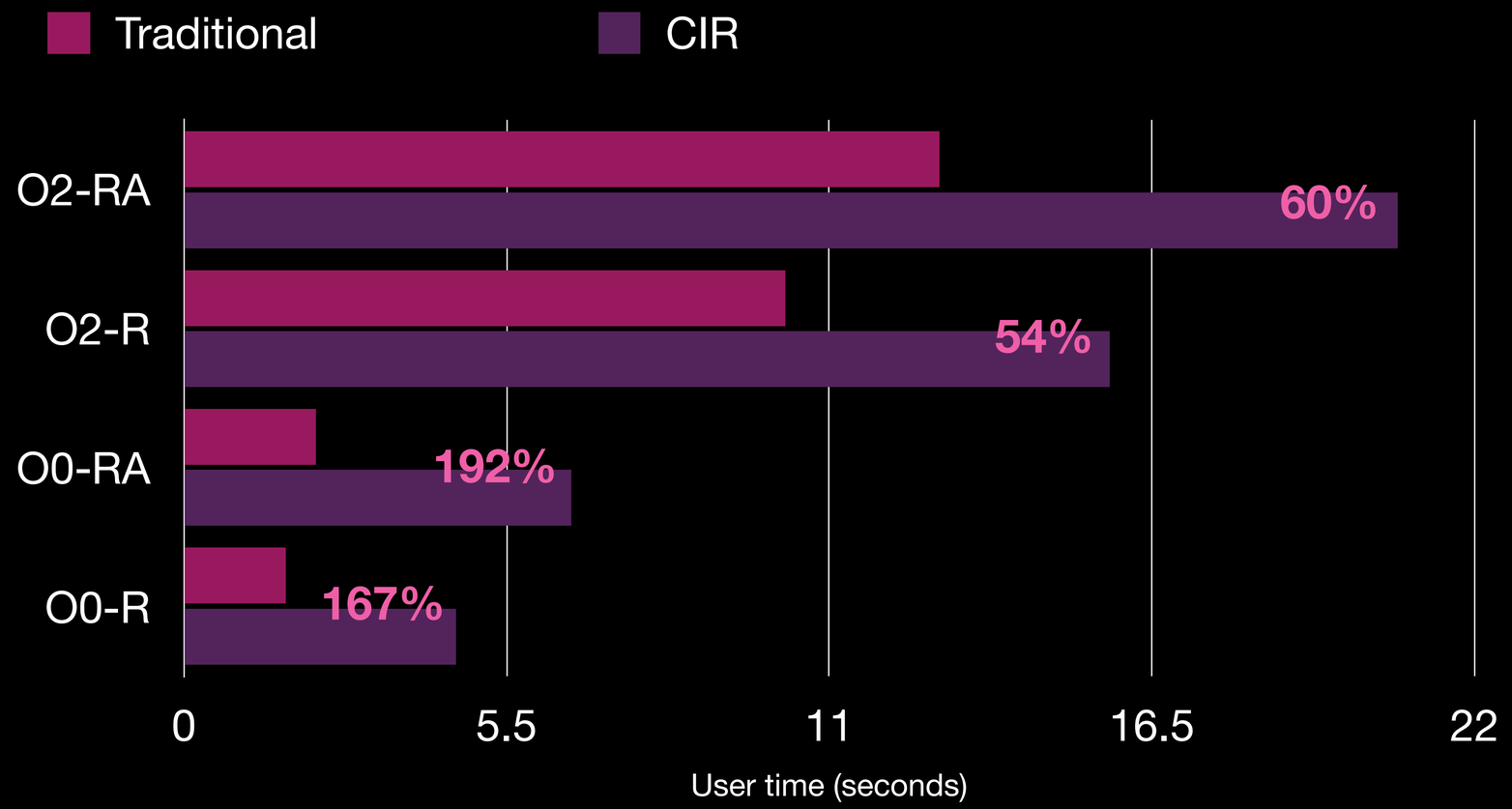
CTMark

SPASS (50 C sources files)



CTMark

lencod (54 C source files)



Multisource case study

clamr_cpuonly.cpp

- *-O0 Release*: 53% increase
  - 0.57s to 0.88s
  - Traditional CodeGen: 15ms
  - ClangIR total: 333ms

Multisource case study

clamr_cpuonly.cpp

- *-O0 Release*: 53% increase
  - 0.57s to 0.88s
  - Traditional CodeGen: 15ms
  - ClangIR total: 333ms

CIR pipeline breakdown

	Time (ms)
CIRGen	289
CIR Verifier	4
CIR to CIR Passes	11
CIR to LLVM	20
LLVM Dialect To LLVM IR	13
Total	337

Multisource case study

clamr_cpuonly.cpp

CIRGen breakdown

	Time (ms)
Emit Deferred	24
Apply Replacements	265
Total	289

Multisource case study

clamr_cpuonly.cpp

CIRGen breakdown

	Time (ms)
Emit Deferred	24
Apply Replacements	265
Total	289

`CIRGenModule::applyReplacements()`

- `replacePointerTypeArgs (50%)`
- `replaceAllSymbolUses (50%)`

Dtor/ctor completion and replacement by aliasee's

Needs investigation, low hanging fruit

Multisource (overall)

Individual files *-O2 Release*
Compile time > 0.5s

Slowdown	Number of sources
10-20%	20
20-30%	30
30%-above	7

```
lencod/rdopt.c 308.5%
  lemon/lemon.c 91.8%
consumer-typeset/z08.c 75.3%
  ldecod/mbuffer.c 51.0%
lencod/macroblock.c 38.9%
ldecod/macroblock.c 36.2%
  Alencod/mbuffer.c 34.9%
  lencod/me_epzs.c 33.2%
Symbolics-flt/tsc.c 28.1%
```

Conclusion

- Opportunity for improvements, low hanging fruits
- Future work
  - More C++ support, analyze other codebases
  - New CIR passes and optimizations
  - Follow good practices and avoid anti-patterns
    - *How Slow is MLIR?* talk — EuroLLVM 24'