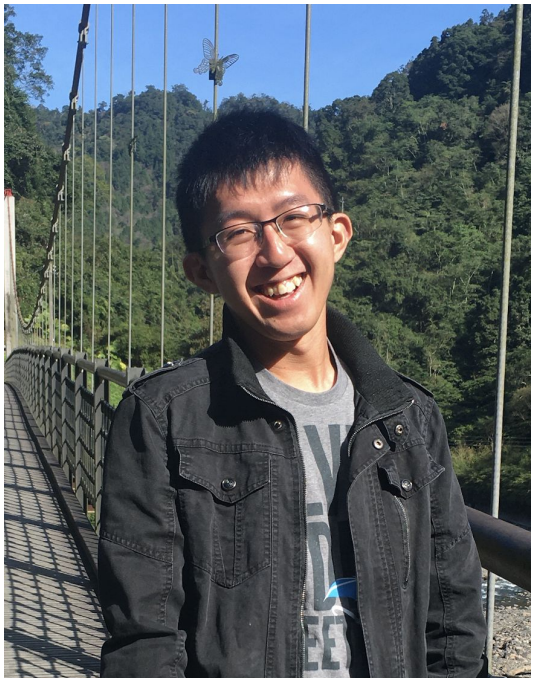
LLVM Backend for M68K

Overview & Status Update

Min-Yih Hsu <minyihh@uci.edu>

Who am I

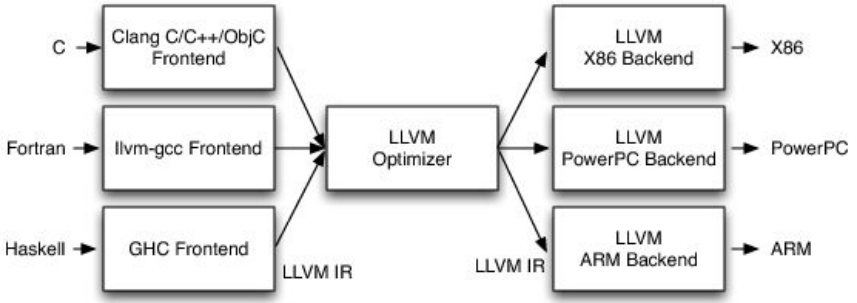


- Min-Yih “Min” Hsu
- PhD Student @ UC Irvine, CA, USA
- Research Interests: Compiler, System Architecture and System Security
- Hanging out with LLVM for ~4 years
- A coffee junkies, chocolate lover, and motorcycle rider

Background

~~Introduction to LLVM~~ LLVM/Clang Highlights

- Native support for cross compilation
- Built-in assembler (and linker)
  - Now LLVM even has its own libc
- High quality code base supported by major tech giants
- Clang and libcxx keep really closed to the C/C++ standard

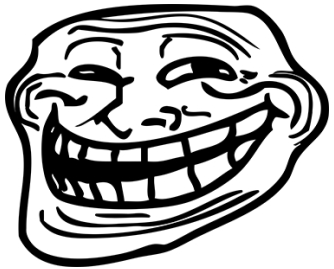


History of LLVM Backend for M68K

Adrian

Artyom

- Feature Complete!!
- With integration tests
- Based on LLVM 8.0
- Made many changes on target-independent parts

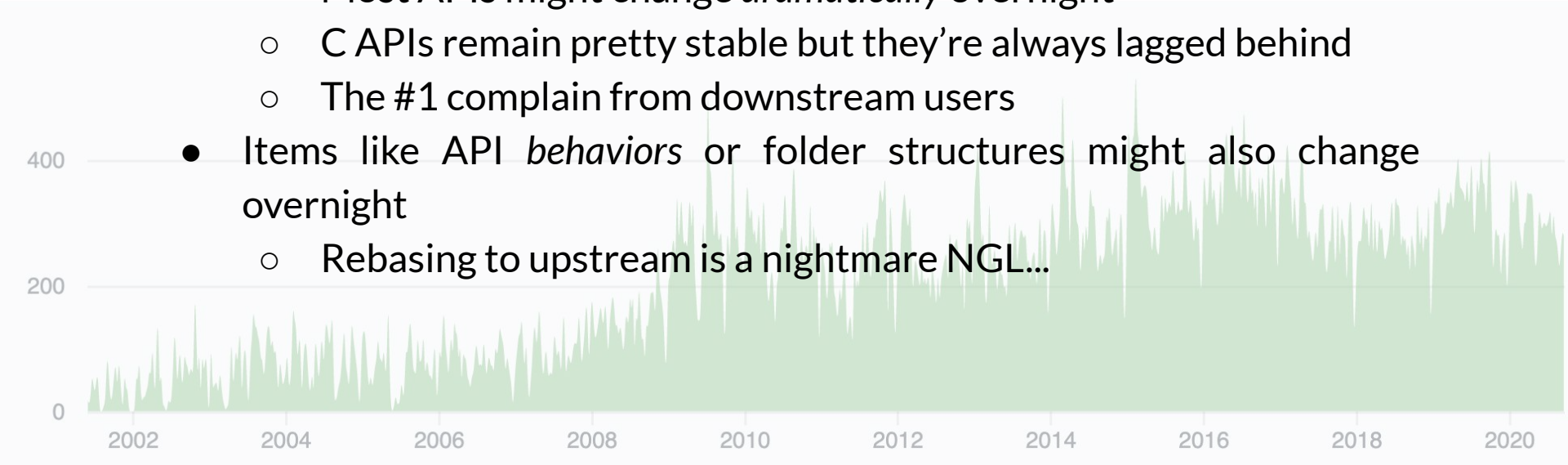


Me

Challenges

The Fast-Paced LLVM

- Thousands of contributors w/ dozens of commits every hour
- LLVM has **no guarantee** on C++ APIs stability
  - Most APIs might change *dramatically* overnight
  - C APIs remain pretty stable but they're always lagged behind
  - The #1 complain from downstream users
- Items like API *behaviors* or folder structures might also change overnight
  - Rebasing to upstream is a nightmare NGL...



LLVM Backends are Complicate

It’s not just code generations…

- Legalizations
  - “Legalize” instructions
  - Frame lowering
  - Calling conventions
- Instruction selections & scheduling (can be automated by TableGen)
- The MC layer (a.k.a Assembler framework)
  - Assembly printing
  - Object file generations

LLVM Backends are Complicate

The X86 backend, being obscured & complex AF

Progress

`clang -triple m680x0 -c hello.c`

Functional Clang @ **tip-of-tree** LLVM

~90% Test Passing Rate

68 out of 77 integration tests have passed

- Support different sub-architectures
  - Mx68000 ~ Mx68040
- Support ISR (Interrupt Service Routine)
- Currently only support ELF Linux ABI

Future Plan

- Aiming for 100% test passing rate!
- Tested with more real-world applications
- LLD (i.e. LLVM’s linker) support
- Tested with libcxx / LLVM’s libc
- Adopting the new instruction selection framework

Upstream process kicked off!

[llvm-dev] [RFC] Backend for Motorola 6800 series CPU (M68k)

Min-Yih Hsu via llvm-dev llvm-dev at lists.llvm.org
Thu Sep 24 16:31:25 PDT 2020

- Previous message: [llvm-dev] New TableGen documents
- Next message: [llvm-dev] [RFC] Backend for Motorola 6800 series CPU (M68k)
- **Messages sorted by:** [date] [thread] [subject] [author]

```
Hi All,

We would like to contribute our supports for Motorola 68000 series CPU (also known as M68k or M680x0) into LLVM. And we
want to hear feedbacks from you
```

Thank you!

Questions?