facebook
Handling all Facebook requests with JITed C++ code

Yuhan Guo, Huapeng Zhou
Software Engineers, Facebook
User requests

User -> L7 Loadbalancer -> Web servers
User requests

- Core HTTP stack + Business logic
- Large v.s. Small
- Stable v.s. Fast iteration
Other scripting language

- Interfacing ✓
- Testing ★
- Debugging ❌
- Profiling ❌
- Performance ❌
- Side-effects 🛠️
What we want

- Interfacing
- Testing
- Debugging
- Profiling
- Performance
Clang/LLVM
Build time

```cpp
class UrlParser {
    ...
};

int getID(string url) {
    auto p = UrlParser(url);
    ...
}

class UrlParser {
    ...
};

getID(url);
```
Runtime

C++ script

Pre-compiled Header

Clang

LLVM IR

Compiler options from Clang PCH metadata

JIT Engine

Main Binary

Statically linked

Clang/LLVM libraries
Runtime

- C++ script
- Pre-compiled Header

Clang

LLVM IR

Exported Symbols

JIT Engine

Main Binary

Executable code

Clang/LLVM libraries

LLVM

OrcJit

Statically linked
What we want?

- Interfacing
- Testing
- Debugging
- Profiling
- Performance
Interfacing

- C++ script
- Pre-compiled Header
- Main Binary
- LLVM IR
- Exported Symbols
- JIT Engine
- Executable code

Clang

LLVM OrcJit

Statically linked

Clang/LLVM libraries
Interfacing

- C++ script
- Pre-compiled Header
  
  ```c
  while (true) {
    fork();
  }
  ```

- LLVM IR
  
  - Clang
  
  - Exported Symbols
    
    - JIT Engine
    
    - Main Binary
      
      - Statically linked

- Executable code

  - LLVM OrcJit
  
  - `--dynamic-list`

- Clang/LLVM libraries
Testing

Unit test

C++ script

Pre-compiled Header

Test script

Test PCH

JIT Engine

Unit Tester
Testing
Unit test

- Using same JIT Engine
- Google Test
- ASan/UBSan

```cpp
TEST(Foo, bar){
  auto id = getID(url);
  EXPECT_EQUAL(id, 42);
}
```
Testing
Integration test

- Spin up main binary + scripts locally
- Real HTTP test request against local host
Debugging

- Register in-memory symbol files with GDB
- Github JitFromScratch project has been helpful

`llvm::JITEventListener::createGDBRegistrationListener`
Profiling

- *PerfJITEventListener* added in https://reviews.llvm.org/D44892
  - Based on jitmup

- Rolled our own *PerfMapJITEventListener*
  - Based on /tmp/perf-%pid.map

START SIZE symbolname
START SIZE symbolname
...
Performance

Execution time

Time distribution
Cost?

- Addition binary size: ~100MB
- Addition start up time: ~2s
- Quirks: Thread local storage
- Adapt to OrcJIT upstream API change
What’s next?

- Performance tuning based on Perf
- Clang checker
- Module?
- Coroutines-TS?
Thank you!

- [https://llvm.org/docs/tutorial/index.html#building-a-jit-in-llvm](https://llvm.org/docs/tutorial/index.html#building-a-jit-in-llvm) has been extremely helpful
- Continuous support from LLVM society is awesome
- Giving back to community [https://reviews.llvm.org/D53911](https://reviews.llvm.org/D53911)
facebook