
facebook

Handling all Facebook requests with JITed C++ code

Yuhan Guo, Huapeng Zhou

Software Engineers, Facebook

User requests

User requests

• Core HTTP stack	+	Business logic
• Large	v.s.	Small
• Stable	v.s.	Fast iteration

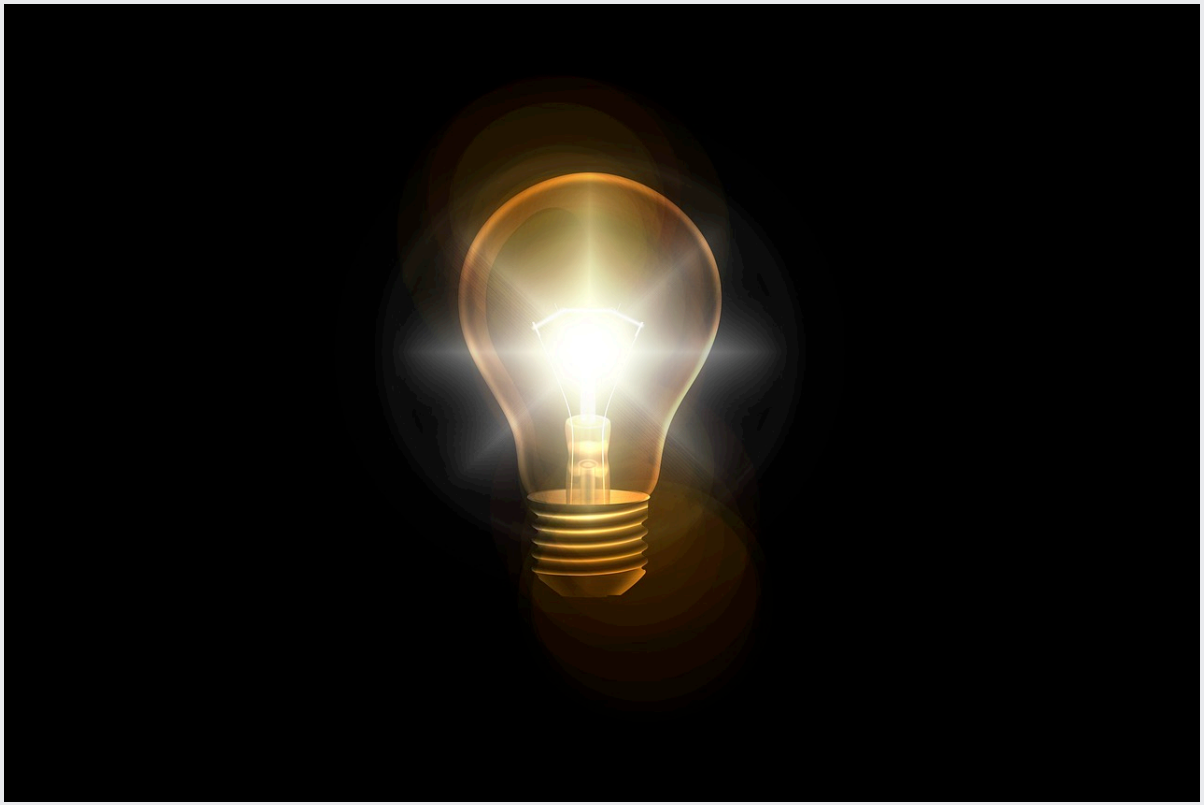
Other scripting language

- Interfacing 
- Testing 
- Debugging 
- Profiling 
- Performance 
- Side-effects 

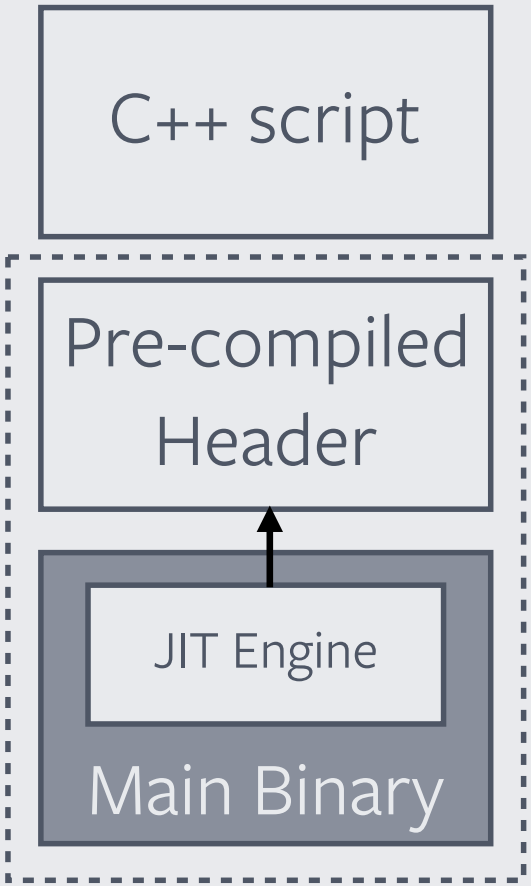
What we want

- Interfacing 
- Testing 
- Debugging 
- Profiling 
- Performance 

Clang/LLVM

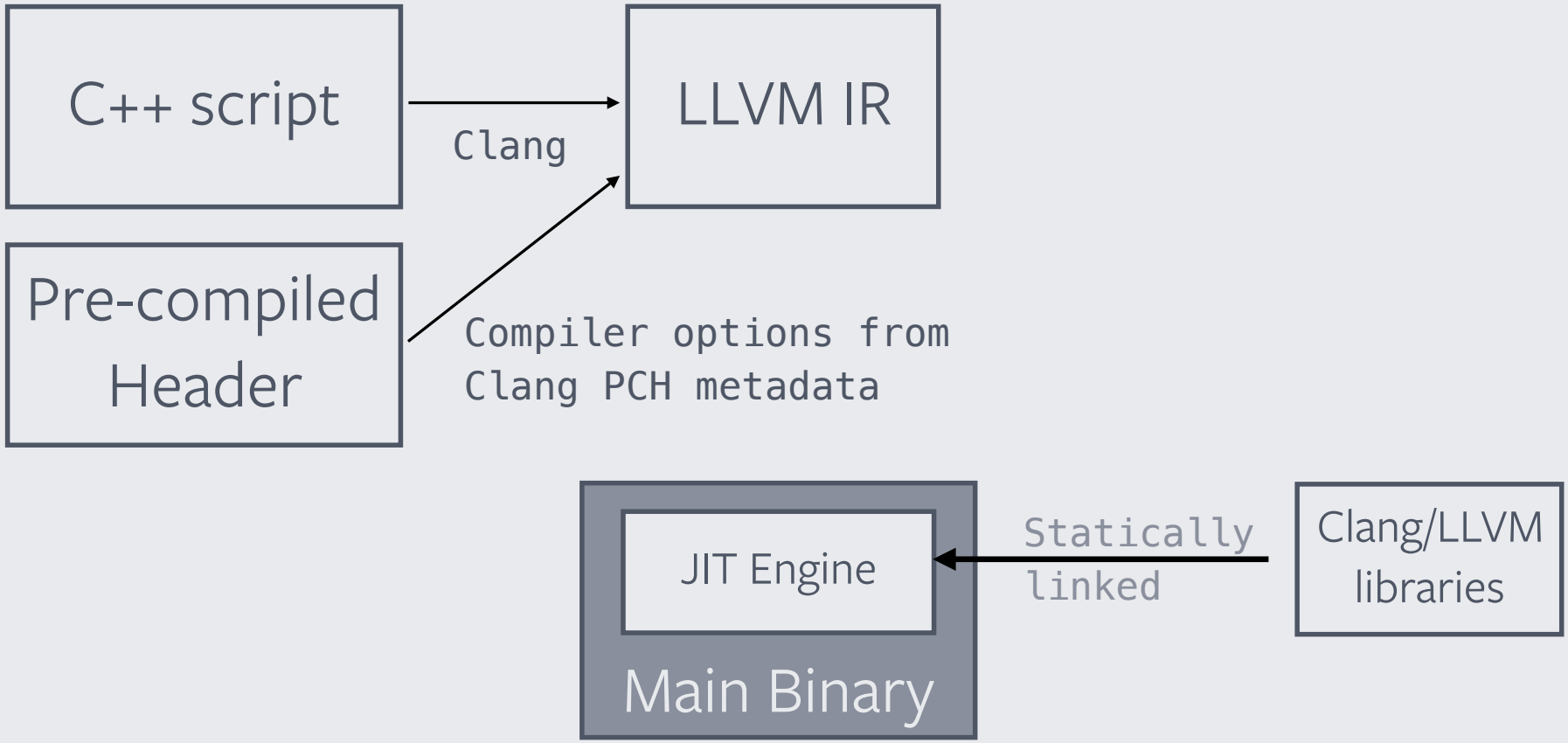


Build time

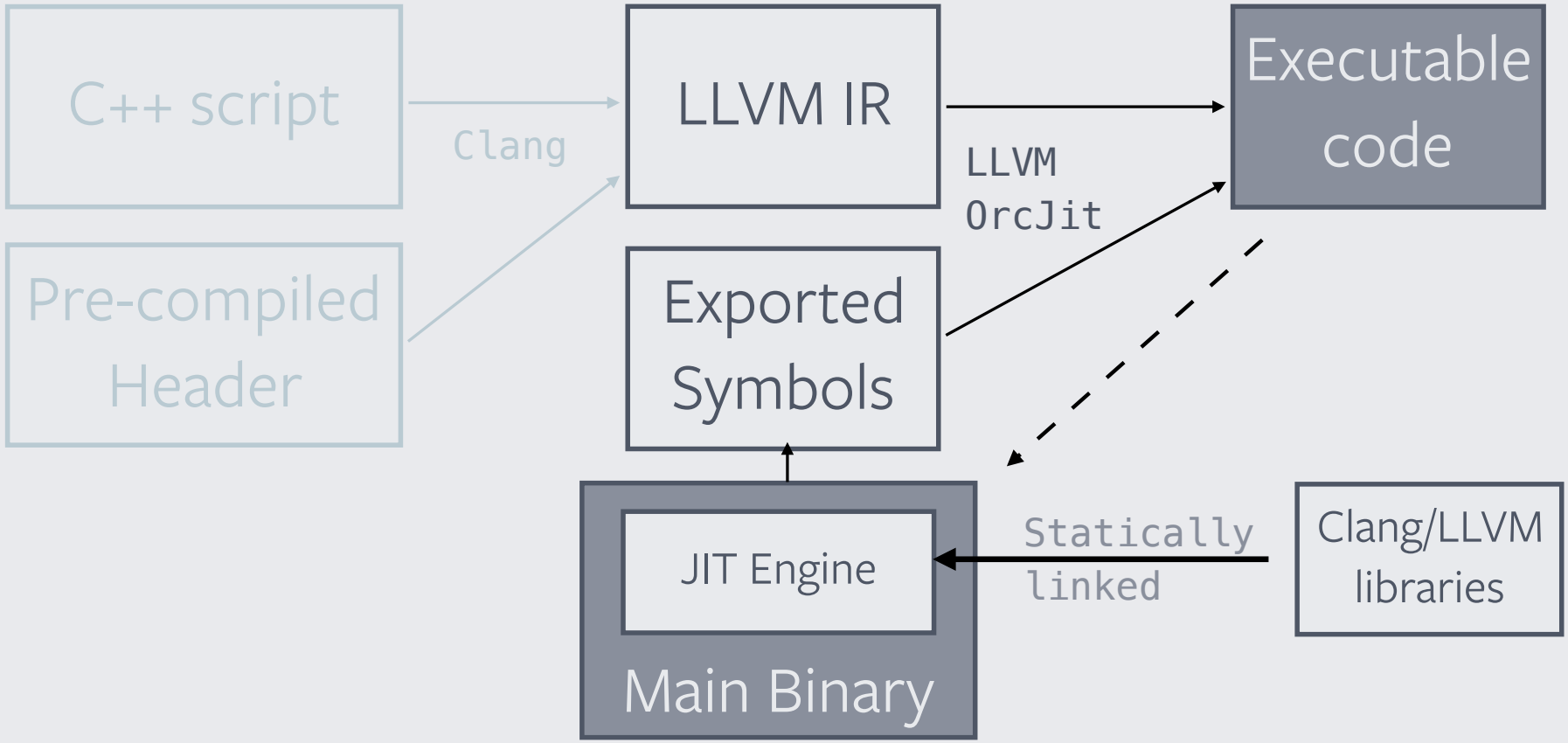



```
int getID(string url) {
  auto p = UrlParser(url);
  ...
}

class UrlParser {
  ...
};

getID(url);
```

Runtime


C++ script
Clang
LLVM IR
Pre-compiled Header
Compiler options from Clang PCH metadata
JIT Engine
Main Binary
Statically linked
Clang/LLVM libraries

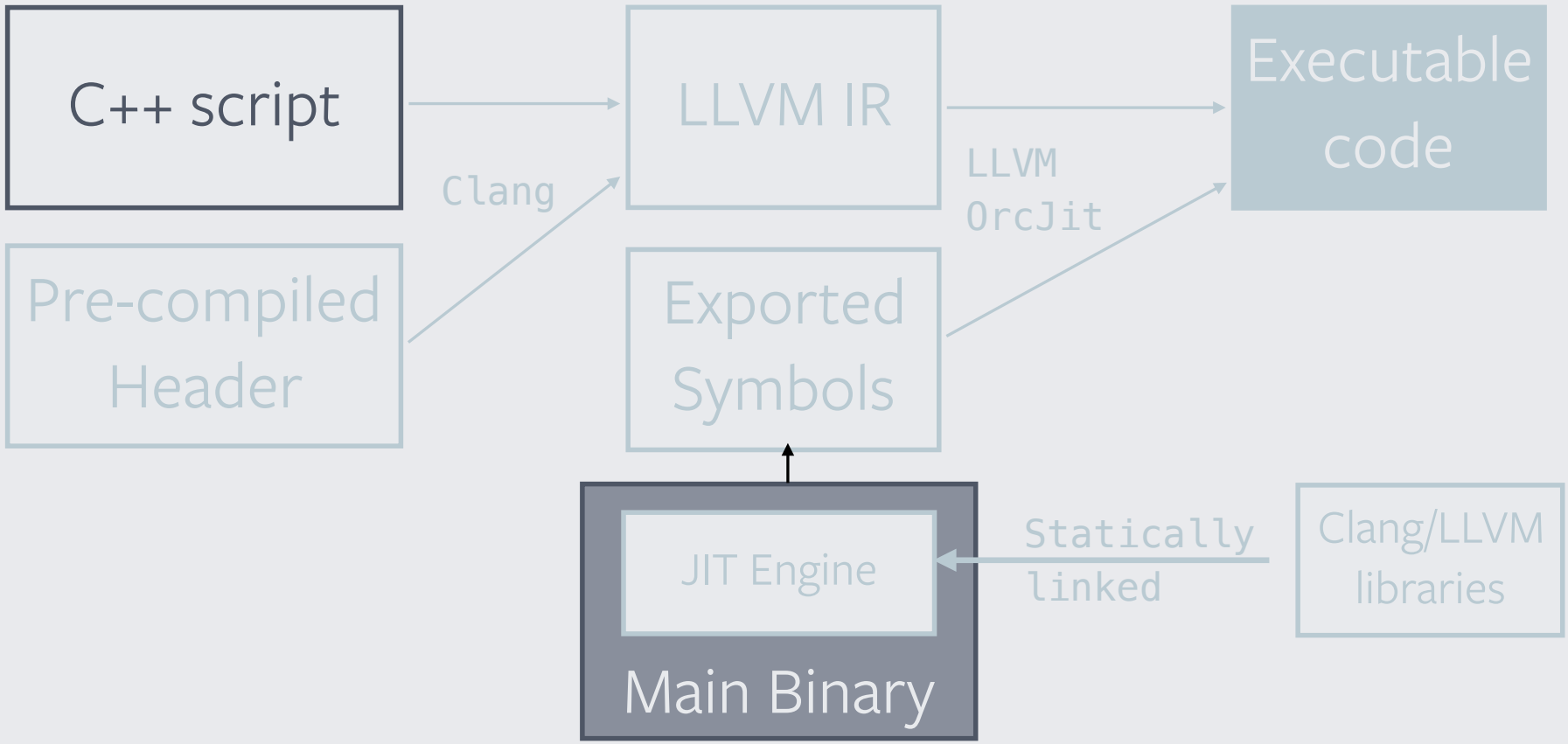
Runtime


C++ script
Pre-compiled Header
Clang
LLVM IR
Exported Symbols
LLVM OrcJit
Executable code
JIT Engine
Main Binary
Statically linked
Clang/LLVM libraries

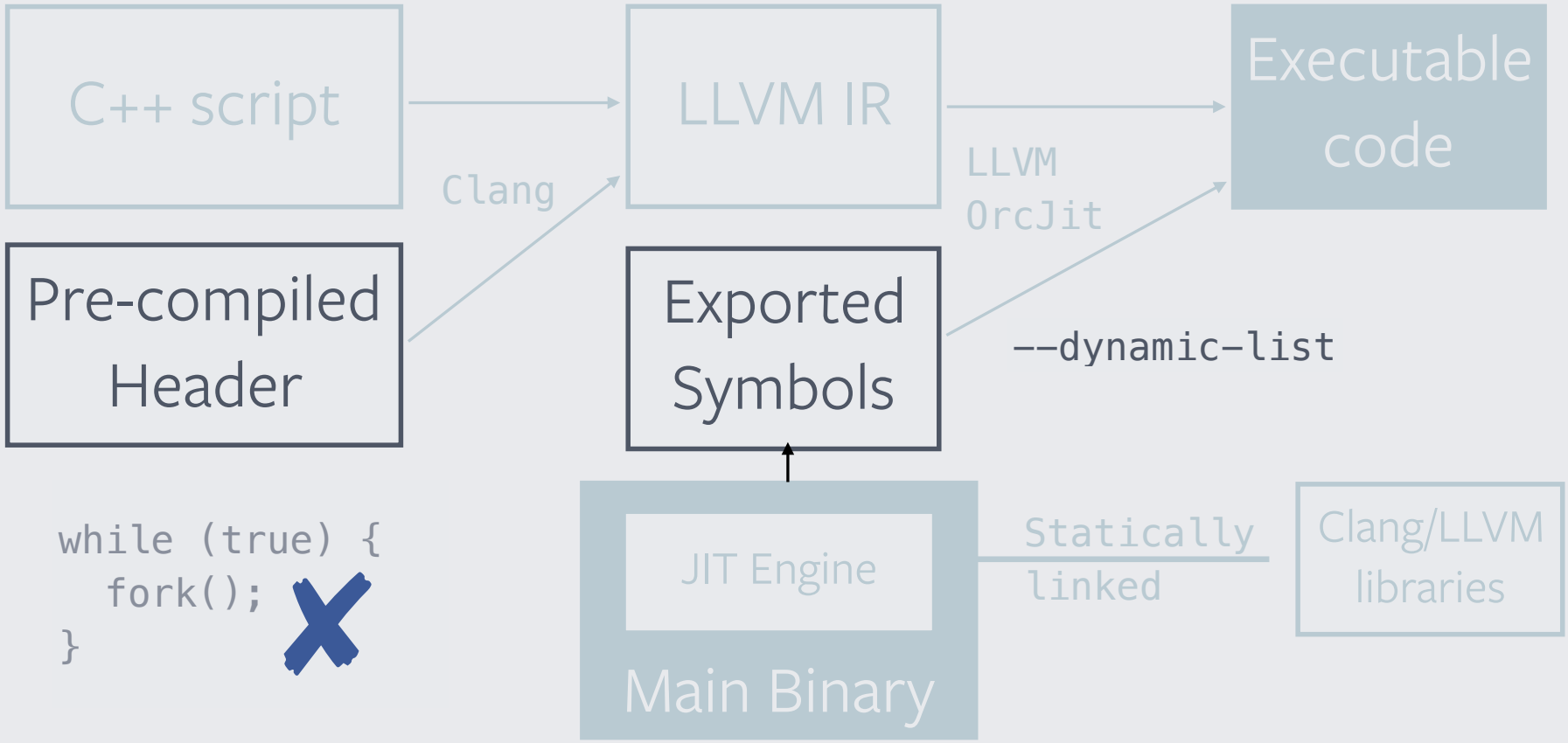
What we want?

- Interfacing 
- Testing 
- Debugging 
- Profiling 
- Performance 

Interfacing

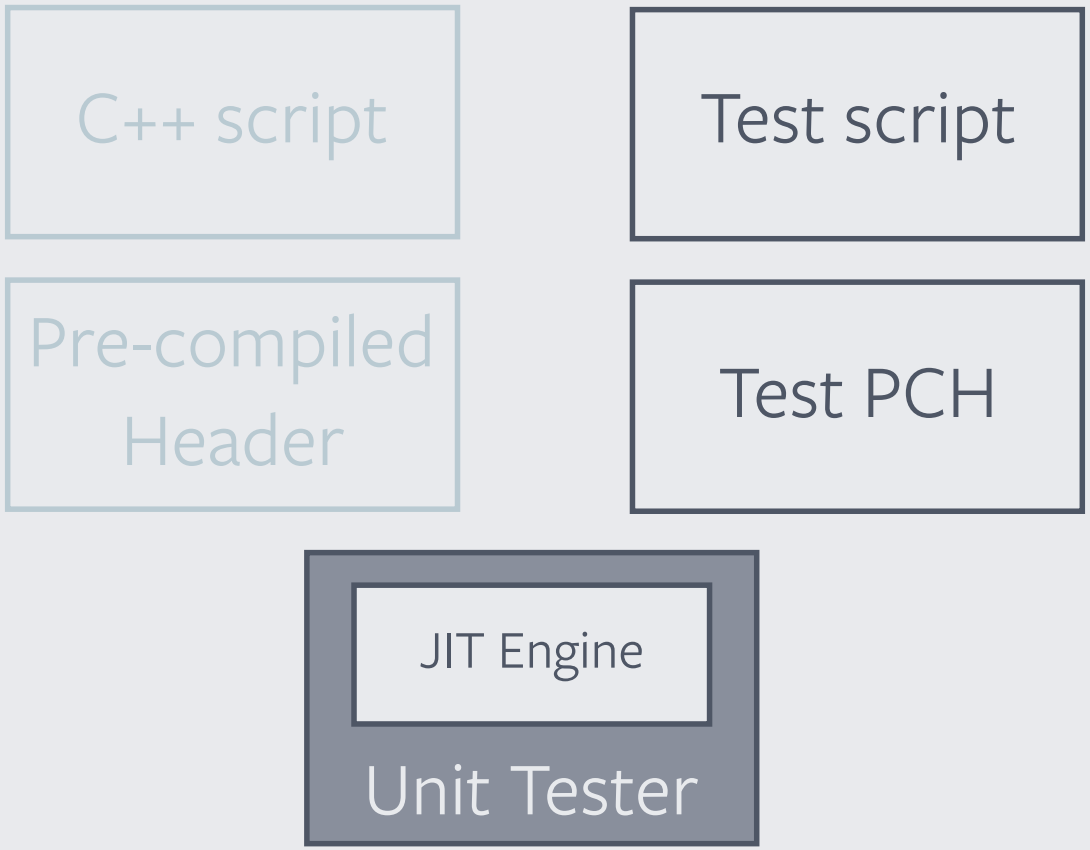


Interfacing



Testing

Unit test



Testing

Unit test

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

```
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
```

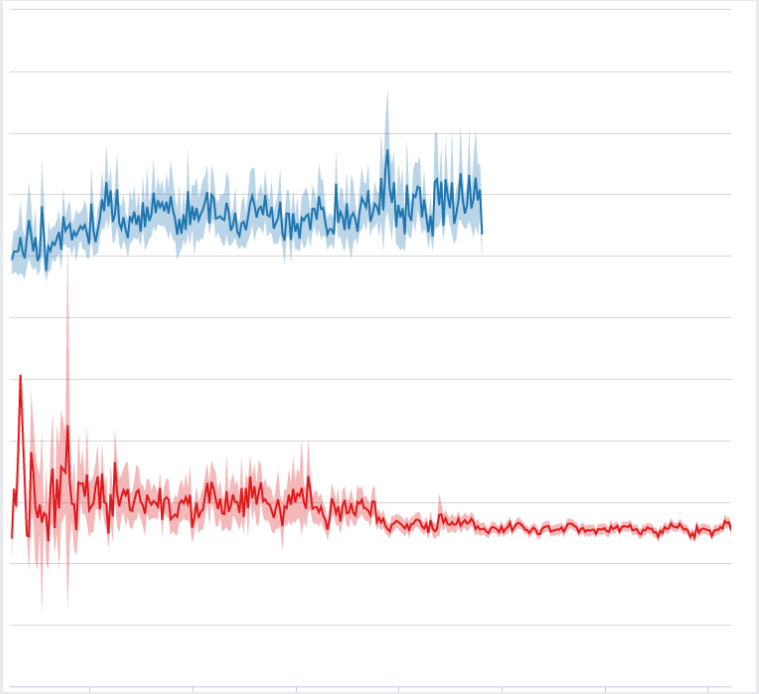
Profiling

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

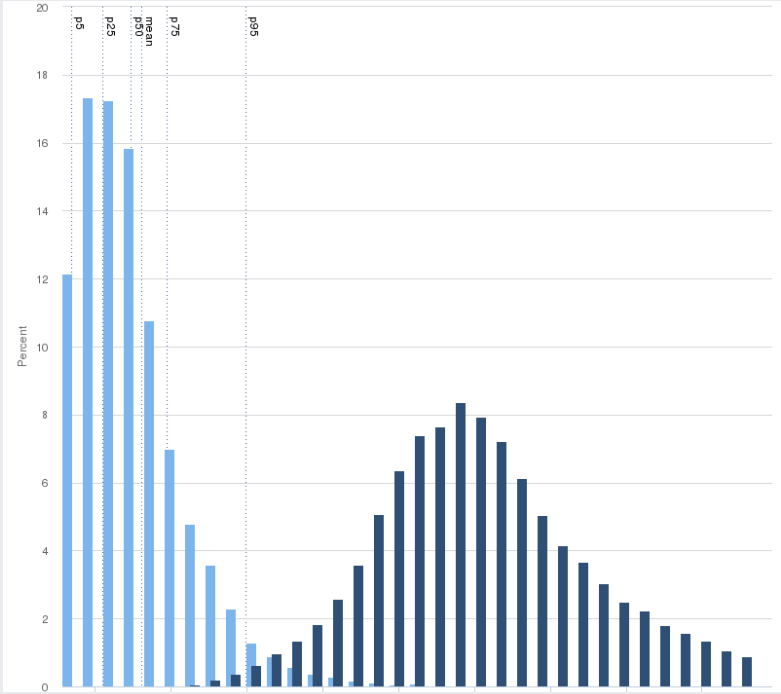
- Rolled our own *PerfMapJITEventListener*
  - Based on /tmp/perf-%pid.map

```
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 has been extremely helpful
- Continuous support from LLVM society is awesome
- Giving back to community https://reviews.llvm.org/D53911


facebook