facebook

Handling all Facebook requests with JITed C++ code

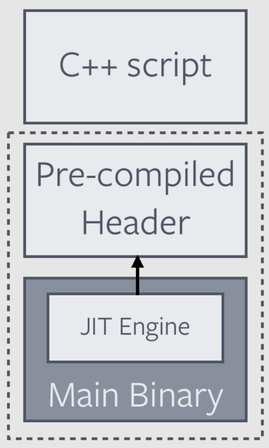
Yuhan Guo(yhguo@fb.com), Huapeng Zhou(hzhou@fb.com)

Background & Motivation

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

	Other scripting language	What we want
Interfacing	✓	★
Testing	★	★
Debugging	✗	★
Profiling	✗	★
Performance	✗	★
Side-effects	☢	

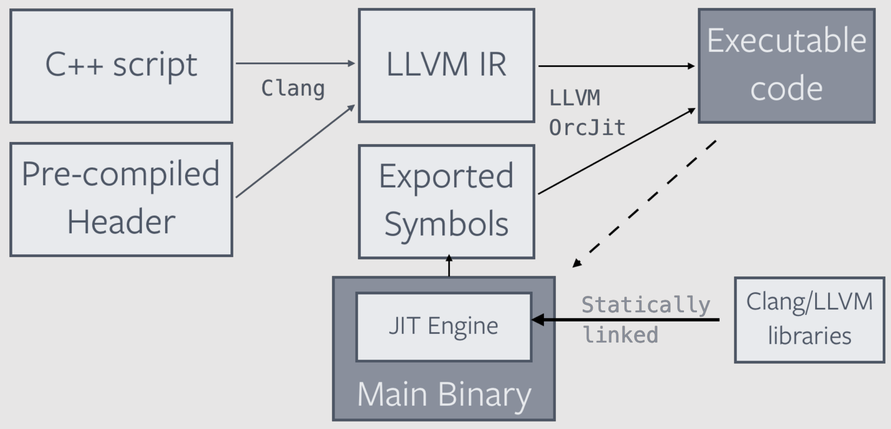
Implementation: Build time v.s. Run time



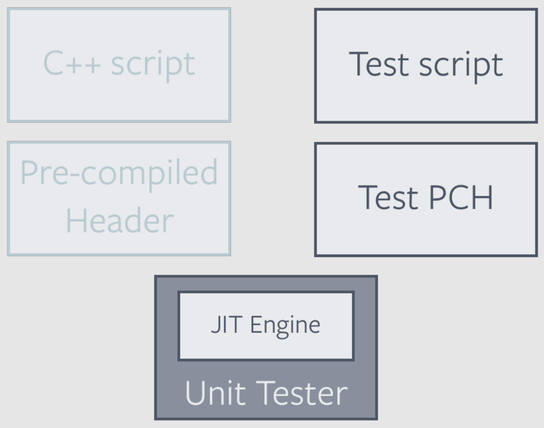

```
int getID(string url) {
  auto p = UrlParser(url);
  ...
}

class UrlParser {
  ...
};


getID(url);
```



Testing




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

Unit test
- Google Test
- ASan + UBSan

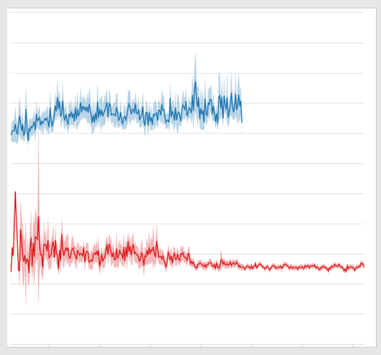
Integration test
- Local instance of main binary + script

Debugging & Profiling

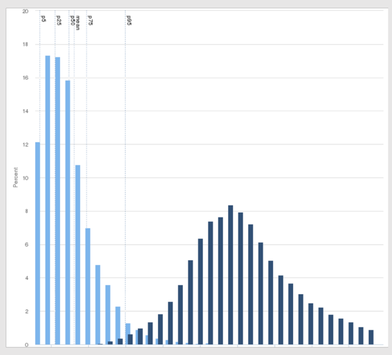
```
llvm::JITEventListener::createGDBRegistrationListener
```

```
/tmp/perf-%pid.map
START SIZE symbolname
...
```

Performance & Cost



Execution time



Time distribution

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

EuroLLVM 2019
Brussels, Belgium