LLVM

——Asa. Ga 2019/07/19

- 简介
- 发展史
- 编译器设计
- 编译步骤
- 实际应用
- WWDC2019新特性

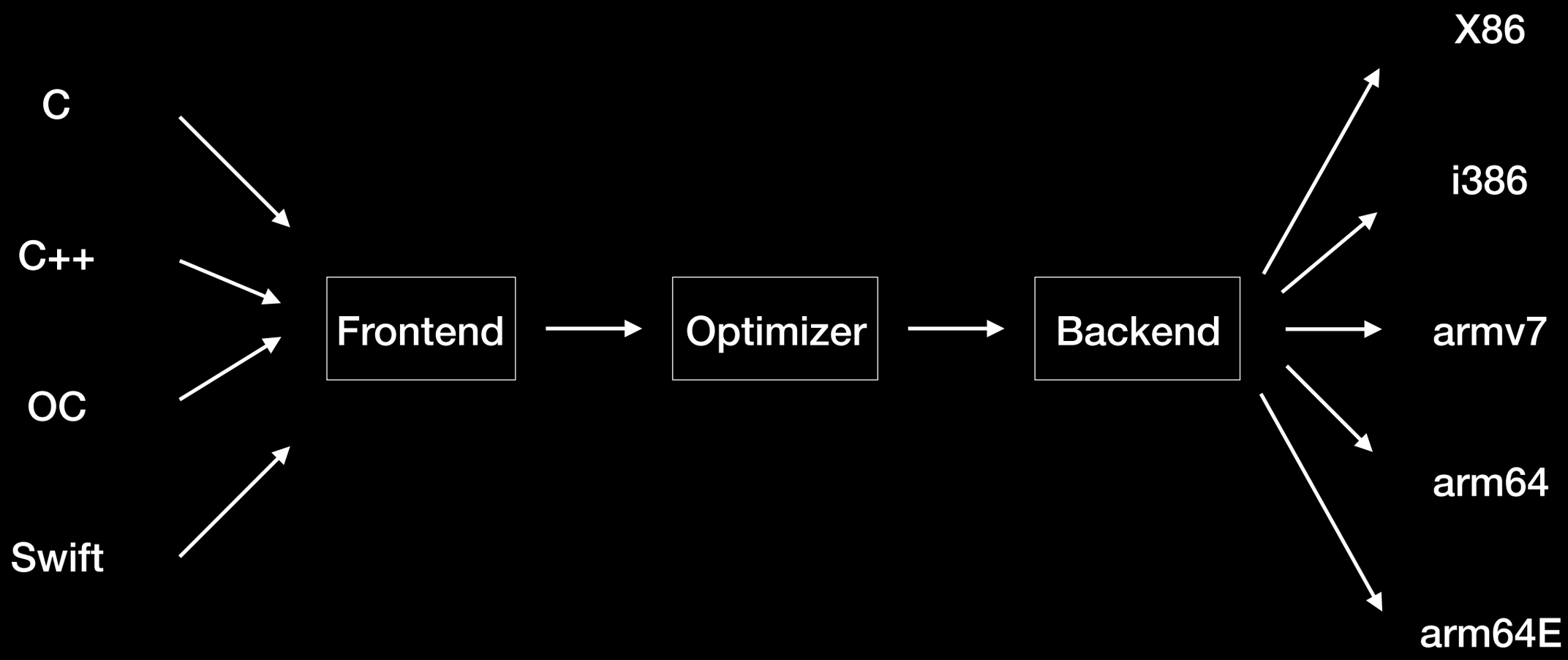
简介

- 编译器：高级语言 ⟶ 机器语言
- LLVM：编译器和工具链技术的集合
- LLVM包含前端、优化器、后端

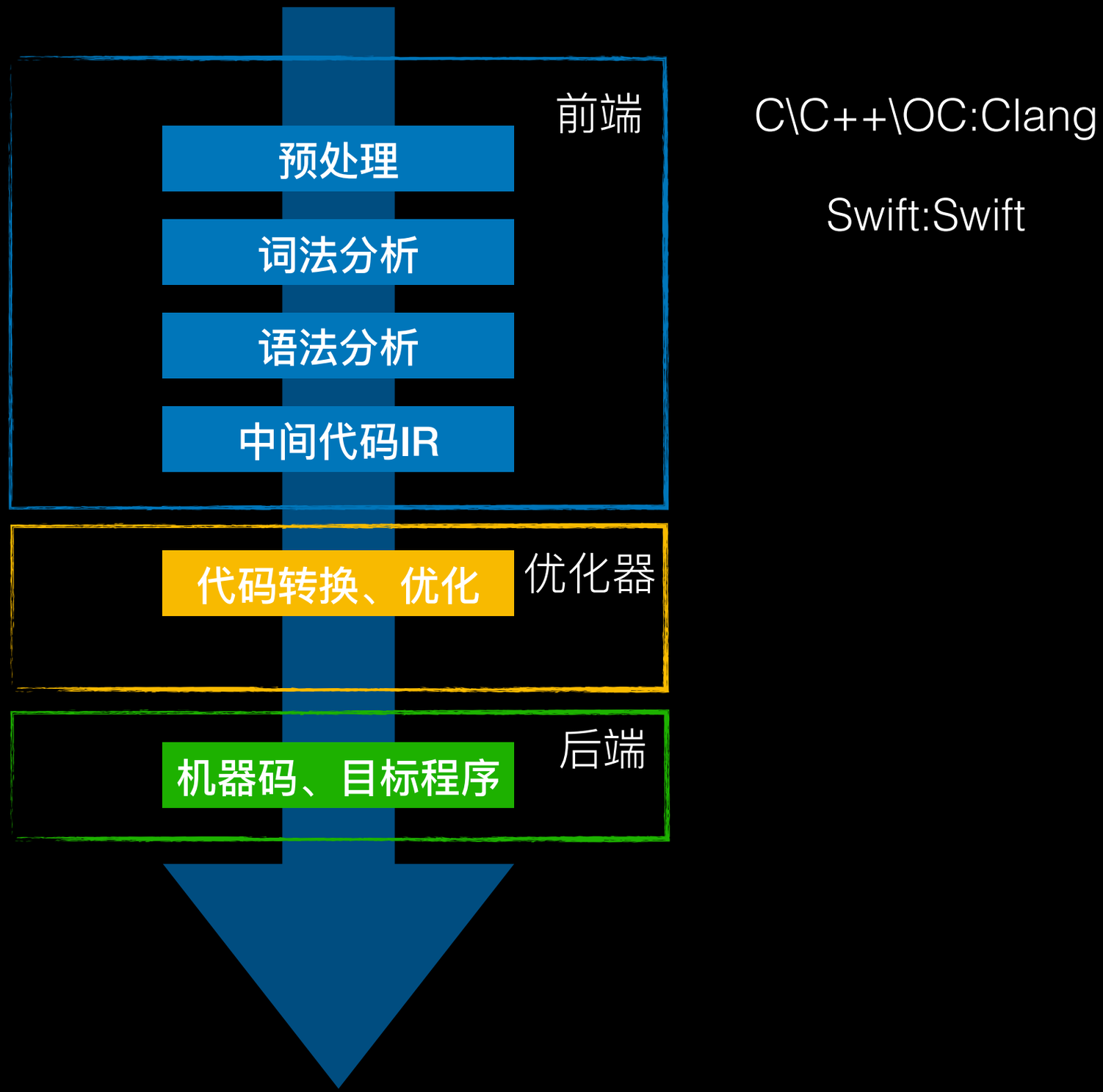
发展史

- Before Xcode3: GCC
- In Xcode3:LLVM,GCC+LLVM
- Xcode4.2:Clang-LLVM 3.0
- Xcode4.6:LLVM 4.2
- Xcode5:abandon GCC, use LLVM5.0

LLVM设计


C
C++
OC
Swift
Frontend
Optimizer
Backend
X86
i386
armv7
arm64
arm64E

编译步骤



C\C++\OC:Clang

Swift:Swift

编译实践

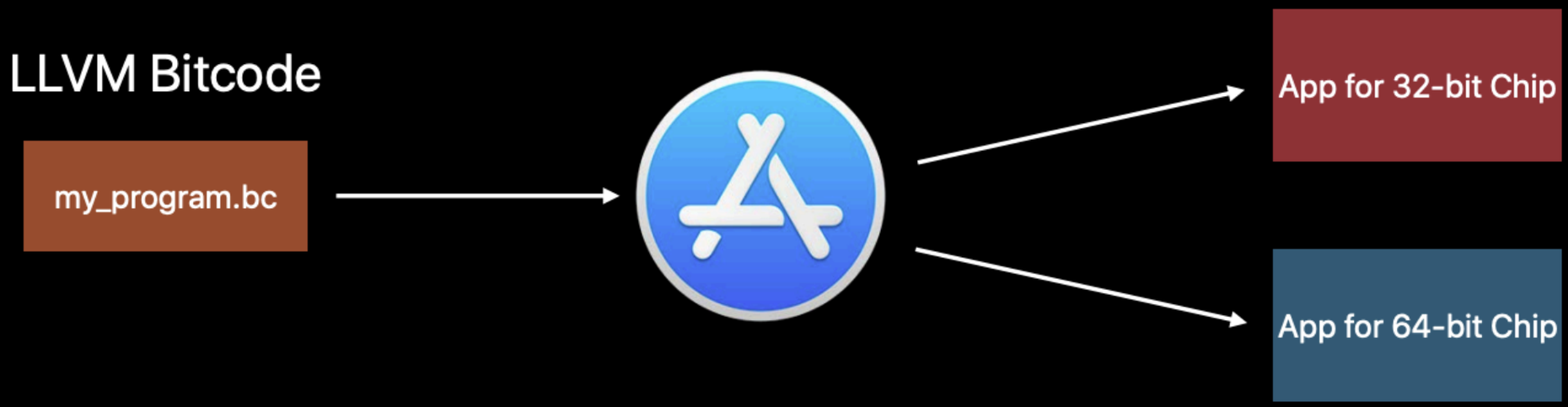
实际应用

- Clang插件：对代码进行个性化检查、命名规范、代码规范
- 语言转化：Object-C转Swift、字符串转OC代码并执行
- 代码混淆

WWDC 2019

- New platform support
- Code Size
- Static Analyzer

Platform


LLVM Bitcode
my_program.bc
App for 32-bit Chip
App for 64-bit Chip

Platform


Bitcode for 32-bit Chip
my_program.bc
App for 32-bit Chip


Bitcode for 64-bit Chip
my_program.bc
App for 64-bit Chip

-Oz

```
hasse:
  …
  ldr   w0, [sp, #16]
  mul   w0, w1, w2
  add   sp, sp, #16
  ret

kakutani:
  …
  ldr   w0, [sp, #16]
  mul   w0, w1, w2
  add   sp, sp, #16
  ret
```

```
OUTLINED_FUNCTION_0:
  ldr   w0, [sp, #16]
  mul   w0, w1, w2
  add   sp, sp, #16
  ret
```

-Oz

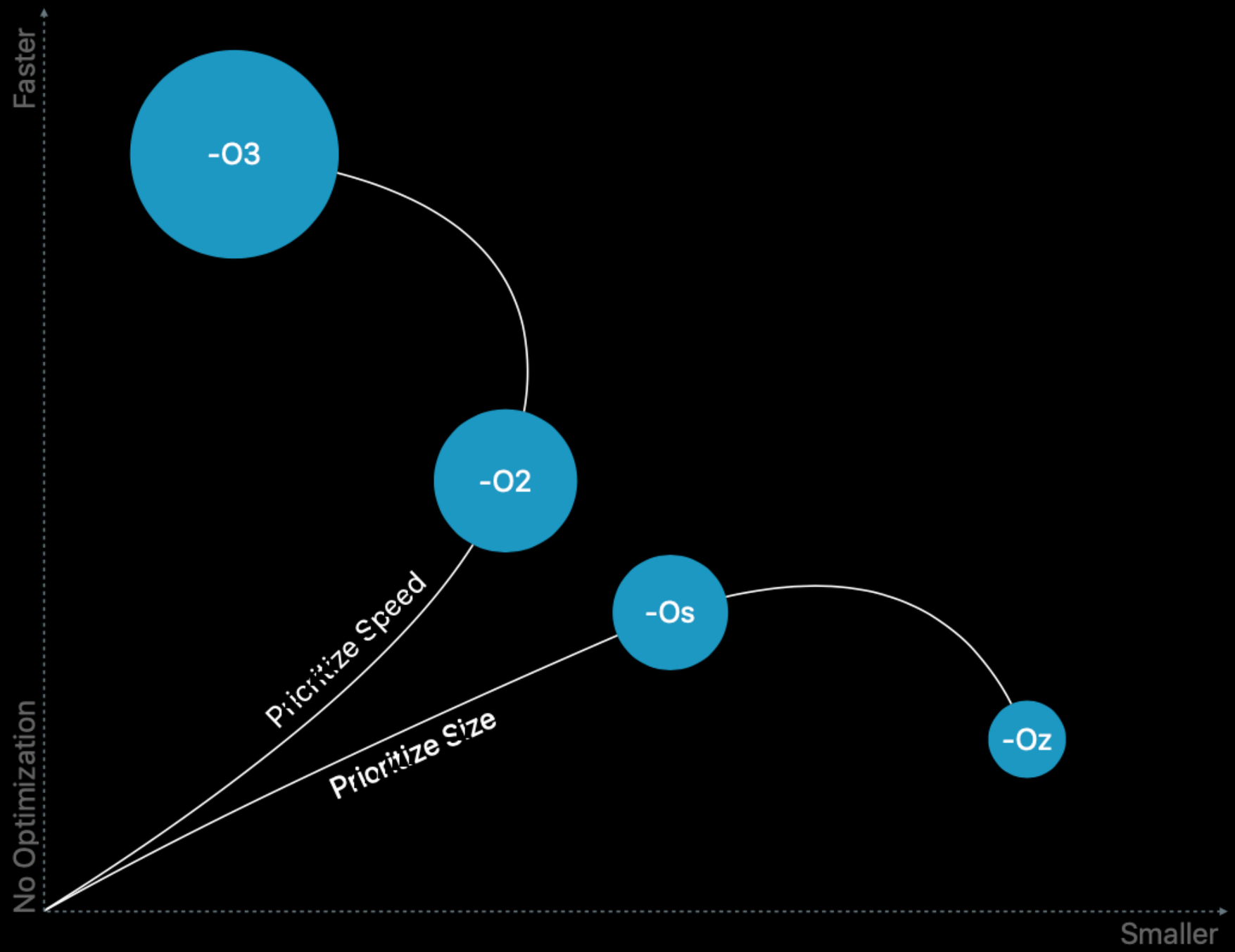
```
hasse:
  …
  b OUTLINED_FUNCTION_0

kakutani:
  …
  b OUTLINED_FUNCTION_0
```

```
OUTLINED_FUNCTION_0:
  ldr   w0, [sp, #16]
  mul   w0, w1, w2
  add   sp, sp, #16
  ret
```

25%

-Oz


Faster
No Optimization
Smaller
-O3
-O2
-Os
-Oz
Prioritize Speed
Prioritize Size

-Oz


Compile Sources (3 items)
Name
Compiler Flags
ViewController.m ...in Sniffo
-O2
main.m ...in Sniffo
-Oz
AppDelegate.m ...in Sniffo
-Os

Language-level

```
// Generated block metadata - as of Xcode 11

static const char  *__block_method_signature_v16_0_8 = "v16@?0@8";
static const struct { // neutron block metadata
  unsigned long reserved = 0, block_size = 48;
  void         *copy_helper            = ___copy_helper_block_ea8_32s40s;
  void         *destroy_helper         = ___destroy_helper_block_ea8_32s40s;
  const char   *block_method_signature = __block_method_signature_v16_0_8;
  uintptr_t     block_layout_info      = 512;
} ___block_descriptor_48_ea8_32s40s_e8_v16?08l;
static const struct { // proton block metadata
  unsigned long reserved = 0, block_size = 52;
  void         *copy_helper            = ___copy_helper_block_ea8_32s40s;
  void         *destroy_helper         = ___destroy_helper_block_ea8_32s40s;
  const char   *block_method_signature = __block_method_signature_v16_0_8;
  uintptr_t     block_layout_info      = 512;
} ___block_descriptor_52_ea8_32s40s_e8_v16?08l;
```

2%-7%

WWDC 2019

- 低级代码瘦身，引入-Oz：
- 高级语言代码瘦身：block，定义metadata元数据结构

```
struct Metadata {
  unsigned long reserved, block_size;
  void *copy_helper;
  void *destroy_helper;
  const char *block_method_signature;
    uintptr_t block_layout_info;
};
```

- 直接继承*NSObject*类的实例变量访问方式变更。
- 提高*C++*类型的可调式性以及缩减部分代码大小

Static Analyzer

- C++ Move Bugs
- std::string
- DriveKit/IOKit

C++ Move

```
Book myNovel("It was the best of times...");

publish(std::move(myNovel));

myNovel.spellCheck();
```

Method called on moved-from object

std::string

```
const char *generateGreeting(const char *name) {
  std::string greeting = "Hello ";
  greeting.append(name);
  return greeting.c_str();
}

printf("%s from WWDC!", generateGreeting("World"));
```

Returning memory that will be deallocated

Inner string pointer used after deallocation

std::string

```
std::string generateGreeting(const char *name) {
  std::string greeting = "Hello ";
  greeting.append(name);
  return greeting;
}
std::string greeting = generateGreeting("World");
printf("%s from WWDC!", greeting.c_str());
```

DriveKit/IOKit

- 引用计数

命令集

- 查看编译过程：clang -ccc-print-phases
- 预处理: clang -E
- 词法分析:
  clang -fmodules -fsyntax-only -Xclang -dump-tokens main.c
- 语法分析：
  clang -fmodules -fsyntax-only -Xclang -ast-dump main.c
- 生成中间代码：clang -S -emit-llvm main.c
- 生成汇编:clang -S -fobjc-arc main.c -o main.s
- 生成目标文件：clang -fmodules -c main.c -o main.o
- 生成可执行文件：clang main.o -o main
- 执行：./main

参考资料

- http://llvm.org/
- https://developer.apple.com/videos/play/wwdc2019/409/
- https://github.com/jerryga/llvm_practice

Thanks