



ccache - 让Xcode编译速度飞起来



飞鱼湾 关注

0.144 2017.07.21 14:11:15 字数 1,467 阅读 2,662

今天来介绍一个小工具 **ccache**，其可以提高 **xcode** 的编译速度。说起缘由，是因为我的苹果电脑配置比较低，而每次开发调试或测试打包都需要编译工程，虽然项目工程代码量不算大，但是编译的时间还是很长，尤其是修改了头文件或者 **Archive** 打包时，看着进度条像蜗牛在爬一样，心里都有小动物在奔腾.....

好在遇到了 **ccache**，其原理是通过把项目的源文件用 **ccache** 编译器编译，然后缓存编译生成的信息，从而在下次编译时，利用这个缓存加快编译的速度，目前支持的语言有：**C**、**C++**、**Objective-C**、**Objective-C++**，但是如果找不到 **ccache** 编译器，那么还是会选择 **clang** 编译器来编译源文件。下面介绍一下具体的应用过程：

添加ccache脚本

下面是 **ccache** 的编译脚本，命名为：**ccache-clang**，需要添加到工程根目录，**ccache** 就是通过这个脚本来选择编译器的，该脚本的详细介绍如下：

第1行：指定这是一个 **bash** 脚本

第2行：**type -p ccache** 判断是否存在 **ccache** 命令，**/dev/null** 是空设备文件，**2>&1** 是把标准错误流重定向到标准输出流，所以整行的作用是先判断命令是否存在，如果不存在就把标准错误流重定向成标准输出流后，再重定向到空文件，这样错误就不会在控制台上打印出来了

第3行：设置最大的缓存

第4行：设置源文件的编译模式，默认是 **true**，**ccache** 会先处理源文件，如果缓存没有命中，就编译源文件；而如果是 **false**，就会在缓存没有命中时，不是直接编译源文件，而是编译处理后的文件，但是这可能会有问题，因为不同的编译器，预处理后的文件内容不完全相同

第5行：设置为 **true**，表明 **ccache** 直接利用缓存目录下的文件，而不是拷贝后的文件，这样可以加快编译速度，默认是 **false**

第6行：设置提高缓存命中率，忽略**FILE**、**DATE**、**TIME**等宏，也不检查文件的修改和创建时间是否是最新的等等

第9行：指定日志文件的路径

第10行：执行 **/usr/bin/clang** 命令，**"\$@"** 是指所有参数

第12行：执行系统默认的 **clang** 命令，**"\$@"** 是指所有参数

```
1  #!/bin/sh
2  if type -p ccache >/dev/null 2>&1; then
3      export CCACHE_MAXSIZE=10G
4      export CCACHE_CPP2=true
5      export CCACHE_HARDLINK=true
6      export CCACHE_SLOPPINESS=file_macro,time_macros,include_file_mtime,include_file_ctime
7
8      # 指定日志文件路径到桌面，等下排查集成问题有用，集成成功后删除，否则很占磁盘空间
9      export CCACHE_LOGFILE=~/Desktop/CCache.log
10     exec ccache /usr/bin/clang "$@"
11 else
12     exec clang "$@"
```



飞鱼湾

关注

总资产37 (约3.36元)

IntelliJ IDEA 安装与激活

阅读 252

JNI编程实现（Windows）

阅读 352

推荐阅读

2020年3-5年经验iOS面试总结

阅读 4,335

'2020'iOS面试题大全-点亮你iOS技能树

阅读 3,300

iOS各大厂面试题（Objective-C版本）

阅读 1,329

iOS开发者跟你谈谈面试技术中的坑

阅读 1,970

这一篇百度 iOS 面试总结轰动整个开发界【头条热搜】

阅读 4,203



写下你的评论...

评论3

赞3

...

时，脚本中的 `clang` 也需要修改为 `clang++`，如下：

```
1  #!/bin/sh
2  if type -p ccache >/dev/null 2>&1; then
3      export CCACHE_MAXSIZE=10G
4      export CCACHE_CPP2=true
5      export CCACHE_HARDLINK=true
6      export CCACHE_SLOPPINESS=file_macros,time_macros,include_file_mtime,include_file_ctime,
7
8      # 指定日志文件路径到桌面，等下排查集成问题有用，集成成功后删除，否则很占磁盘空间
9      export CCACHE_LOGFILE='~/Desktop/CCache.log'
10     exec ccache /usr/bin/clang++ "$@"
11 else
12     exec clang++ "$@"
13 fi
```

配置Xcode项目

1. 添加一个CC常量

在项目的 **Build Settings** 里，添加一个常量：`CC`，值为脚本的完整路径：`$(SRCROOT)/ccache-clang` (脚本在根目录下)

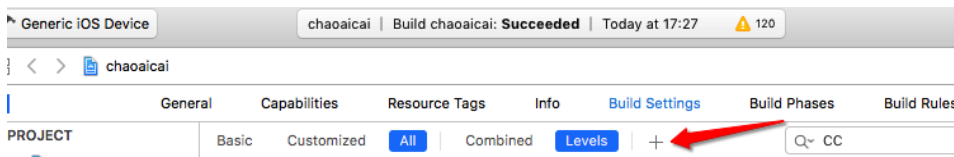


图1

添加 `CC` 常量

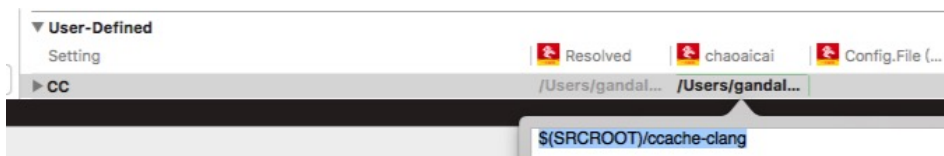


图2

因为 `ccache` 不支持 `Clang Modules`，所以需要关闭项目的 `Modules` 选项，如果你的工程里有用的 `@import UIKit` 等，那么现在需要替换为 `#import <UIKit/UIKit>`，这是因为关闭 `Modules` 后，`Xcode` 不会再自动导入模块了，对于其他的模块，你还需要在 **Linked Frameworks and Libraries** 里手动导入所需模块。

配置Pod项目

如果项目中用到了 `CocoaPods`，还需要设置 `Pod` 工程，因为 `Pod` 工程也需要关闭 `Modules` 选项，我们可以按照修改 `Xcode` 项目的方法来修改 `Pod` 工程的 `modules` 配置，但是每次 `pod update` 会把 `Pod` 工程重新生成，`Modules` 选项又被改回去了，所以我们直接在 `Podfile` 里加入下面的代码，让 `Pod` 工程在生成的时候就关闭 `Modules` 选项，同时加入 `CC` 常量，就可以在编译时用缓存加速了。

但是，由于我们关闭了 `modules` 选项，如果某个 `Pod` 工程用到了系统的框架，则需要按照 `Xcode` 工程的方法添加到对应的 `Pod` 工程里，否则会提示 `Undefined symbols xxx for architecture arm7` 等等，尽管每次更新 `Pod` 后，都需要重新添加所需的框架，但是考虑到编译速度的提升，而 `Pod` 不会经常更新，所以也不是很麻烦。

简书发现关注消息搜索

7

在生成的 Pods 项目文件中加入 CC 参数，路径的值根据你的项目来修改

8

config.build_settings['CC'] = '\$(PODS_ROOT)/ccache-clang'

9

end

10

end

11

end

应用ccache

设置完成后，就可以利用 `ccache` 了，首次因为还没有缓存，要添加缓存，所以编译的时间会比没有用 `ccache` 要多一会儿，但是从第二次之后，会有缓存命中，就可以加快编译速度了，多次之后，可以明显感觉到，编译的速度快了许多，一般至少了加快一倍以上，我们也可以用命令来查看 `ccache -s` 的缓存的命中情况：

```
$ ccache -s
cache directory           /Users/gandalf/.ccache
primary config            /Users/gandalf/.ccache/ccache.conf
secondary config (readonly) /usr/local/Cellar/ccache/3.3.4_1/etc/ccache.conf
cache hit (direct)        2594
cache hit (preprocessed)   844
cache miss                3047
cache hit rate            53.01 %
called for link            17
compile failed            19
no input file              2
cleanups performed        0
files in cache            12108
cache size                412.0 MB
max cache size            10.0 GB
```

图3

可以看到，现在的缓存命中率是 `53.01%`，以前打包需要10分钟，现在3分钟搞定，喝茶的时间又多了，是不是很开心，-。

通过添加 `ccache`，项目的编译速度得到了极大的提升，对于需要花费很长时间进行编译的项目，是十分值得的。

[如何将 iOS 项目的编译速度提高5倍](#)

[ccache — a fast C/C++ compiler cache](#)

[ccache 让你的编译时间飞起来](#)

[linux中shell变量\\$#,\\$@,\\$0,\\$1,\\$2的含义解释](#)

👍 3人点赞 >

👎

📖 日记本

⋮

"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下



飞鱼湾 博主是一名iOS、Java开发者，目前正在朝向全栈的路上潜心修行，希望...
总资产37 (约3.36元) 共写了5.9W字 获得138个赞 共76个粉丝

关注

简书

发现 关注 消息 搜索

Q

Aa beta



数据爬虫II



写下你的评论...

全部评论 3

只看作者

按时间倒序 按时间正序



Corbin__

3楼 2018.06.14 20:37

如果我项目有非常多个target，是不是每个target下也得添加ccache

赞

回复



冰雪飞舞呀

2019.04.04 16:41

多个target 是怎么处理的呢

回复



添加新评论



Corbin__

2楼 2018.06.14 20:33

添加一个CC常量,如果我有ccache-clang++是不是需要再添加一个变量

赞

回复

被以下专题收入，发现更多相似内容

+ 收入我的专题

 iOS plus

推荐阅读

更多精彩内容 >

[贝聊科技]如何将 iOS 项目的编译速度提高5倍

作者 子豪 贝聊iOS工程师 前言 贝聊目前开发的两款App分别是贝聊家长版和贝聊老师版，最近因为在快速迭代开发新...

贝聊 贝聊科技

阅读 2,146

评论 7

赞 20

Android - 收藏集

Android 自定义View的各种姿势1 Activity的显示之ViewRootImpl详解 Activity...

 passiontim

阅读 136,658

评论 20

赞 583

深入剖析 iOS 编译 Clang / LLVM

前言 2000年，伊利诺伊大学厄巴纳－香槟分校（University of Illinois at Urbana-...

 星光社的戴铭

阅读 11,823

评论 7

赞 164

 闲云清烟


阅读 1,833

评论 0

赞 3

shall we dance

Shall we dance 是我将你推进舞池 当迷雾升起 眩目的灯光迷惑了双眼 当脚步开始挪动 不止是相随的身影...

 潜水的鸟

阅读 54

评论 0

赞 2

写下你的评论...

 评论3

 赞3

