

# 技术分享

薛运成



Go build

Concurrency

03 Dead lock

### Go build

编译快的原因

#### **#1 Language**

Generics不支持template

#### #2 Toolchain

不使用LLVM编译库,虽然很强大,但是很复杂,有时候还有点慢

#### **#3 Dependency**

MVS(Minimal Version Selection)算法很简单,方便找到module。

#### **#4 Caching**

做一件事情最快的方式是一点都不做,复用之前做过的。

#### #4a Module Caching(GOMODCACHE)

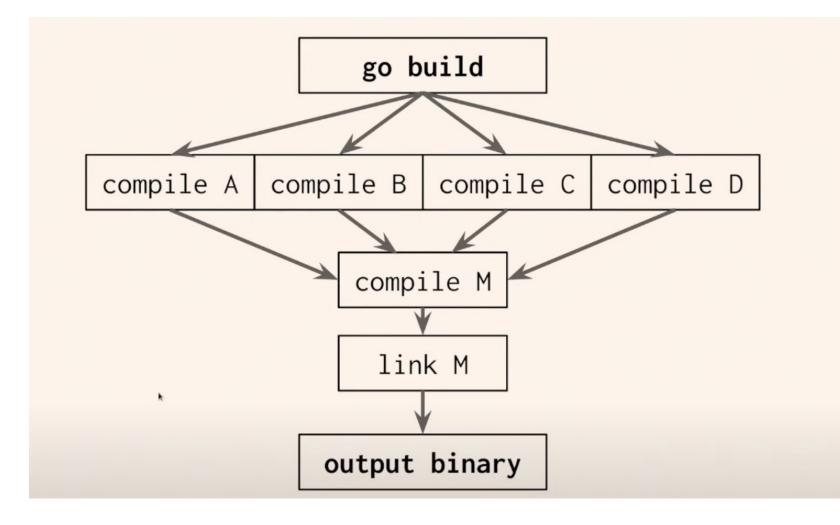
- 1. \$(go env GOROOT)/src/{{PATH}}
- 2. \$(go env GOPATH)/src/{{PATH}}}
- 3. go get 下载包,然后放到GOPATH路径下

#4b Build Caching(GOCACHE)

#### **#5 Parallelism**

```
$ cat main.go
package main // "M"

import (
    "A"
    "B"
    "C"
    "D"
)
```



#### **#6 Reproducibility**

- Build ID, 相同的输入会产生相同的输出。(GODEBUG=gocachehash=1)
- 编译环境一致
  - 编译时使用-trimpath参数
  - 记录go的版本
  - 记录编译时的所有选项
  - 使用CGO\_ENABLED=0

#### **#6 Supply Chain Security**

go.sum文件中包含使用库的hash值

## 02 Concurrency

并发

#### Concurrency

#### demo

问题:给定一个网站列表文件,返回所有NameServer为cloudflare的网站

https://github.com/xueyuncheng/talk2023/tree/master/concurrency

## 03 Dead lock

死锁

### **Dead lock**

#### demo

https://github.com/xueyuncheng/talk2023/tree/master/deadlock

#### 参考链接

<u>Deep dive into Go's build system - Daniel Marti</u>

# Thanks