



#SECONRU



МЕЖРЕГИОНАЛЬНАЯ КОНФЕРЕНЦИЯ
РАЗРАБОТЧИКОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Go на практике



Цаль-Цалко Иван





Почему выбрали Go?

- статически типизированный язык
- высокая производительность
- ориентирован на клиент-серверное взаимодействие
- один бинарный файл для деплоя
- разработчик Google



- сохранение прогресса игрока
- битва гильдий с боссами
- битва PVP
- чат
- нотификации

Мир игрока



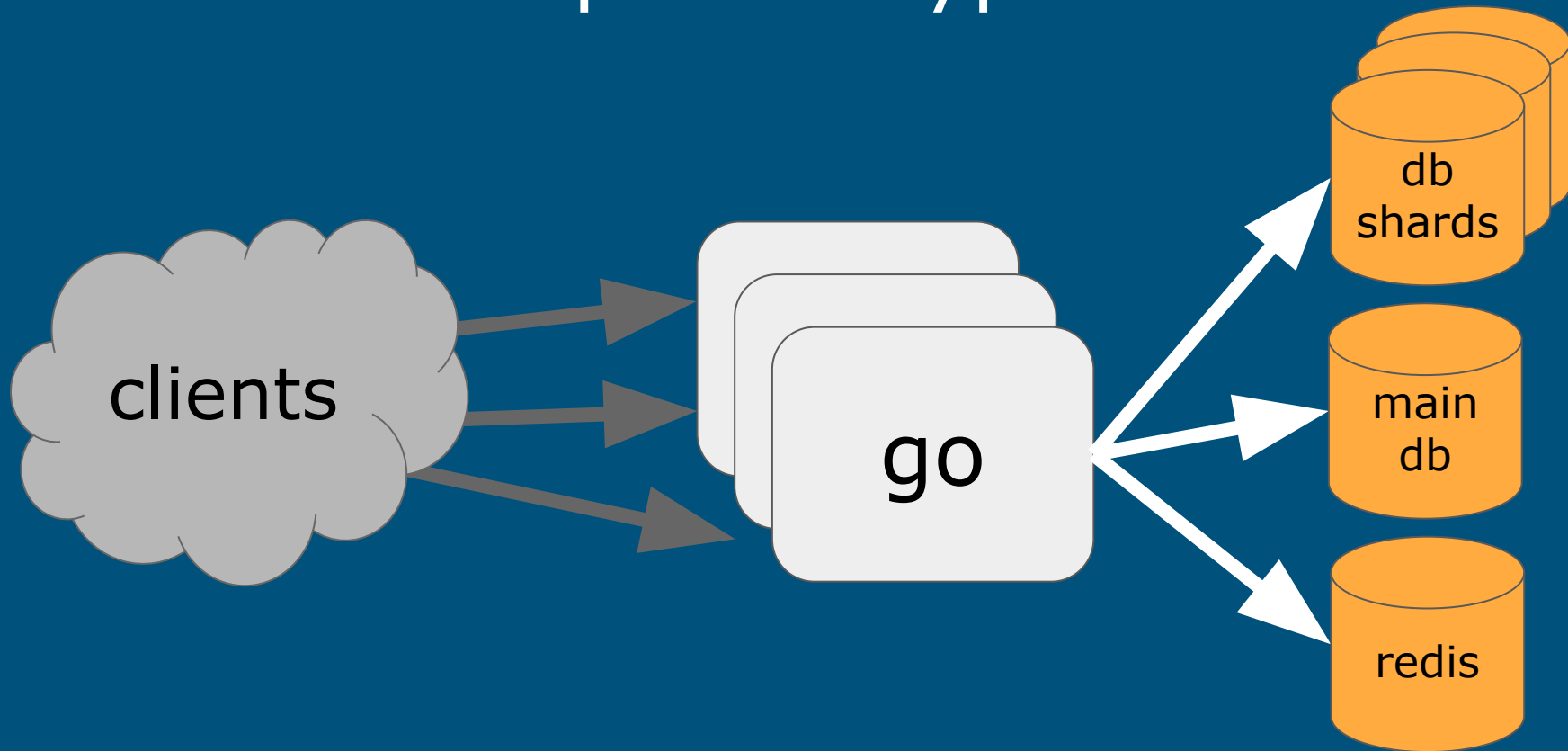
Битва РvР



Битва с боссом



Архитектура



Обработка запросов

- net/http

```
func (*rpc) ServeHTTP(w http.ResponseWriter, r *http.Request) {  
    ...  
    resp := handleRpc(r)  
    w.Write(resp)  
}
```

Особенности языка

- запуск отдельных го-рутин с помощью **go**

```
func (rpc *Rpc) loadWorld(plr_id uint32) (error, DataWorld) {  
  
    player := rpc.loadPlayer(plr_id)  
    go sendStat(EnumStatOnline, player.Id)  
  
    // continue loading world  
    ...  
}
```

Особенности языка

- передача **нескольких** значений в качестве результата

```
func (c ConfigMgr) getById(id uint32) (error, Config) {  
    config, ok := c.id2conf[id]  
    if ok {  
        return nil, config  
    }  
    cnfErr := fmt.Errorf("Not found config by id: %d", id)  
    return cnfErr, nil  
}
```

Особенности языка

- обработка критических ошибок (**panic**)

```
func getOnlinePlayers(req Request) ([]byte, error) {  
    players, err := loadOnlinePlayers(req)  
    CheckErr(err)  
    ...  
}
```

```
func CheckErr(err error) {  
    if err != nil {  
        panic(err)  
    }  
}
```

Особенности языка

- обработка критических ошибок (**panic**)

```
func (*rpc) handleRpc(req *http.Request) ([]byte, error) {  
    defer func() {  
        if err := recover(); err != nil {  
            ...  
        }  
    }()  
    ...  
    players, err := getOnlinePlayers(req)  
    ...  
}
```

Особенности языка

- функция **defer**

```
func GetExtIdById(id uint32) string {  
    conn := GetNewMainConn()  
    defer conn.Db.Close()  
  
    shard_map := GetShardById(conn, id)  
    return shard_map.Ext_id  
}
```

Автогенерация кода

Data.meta:

struct DataWorld

```
player      : DataPlayer
items       : DataItem[]
regions     : DataRegion[]
dungeons    : DataDungeon[]
boosters    : DataBooster[]
```

...

end

struct DataPlayer

```
@table:player @pkey:id @owner:id
id            : uint32
level        : uint32
reg_time     : uint32 @default:0
name         : string @default:""
```

...

end

data.go:

type DataWorld struct {

```
Player      DataPlayer
Items       []*DataItem
Regions     []*DataRegion
Dungeons    []*DataDungeon
Boosters    []*DataBooster
```

...

}

type DataPlayer struct {

```
Id          uint32
Level       uint32
Reg_time    uint32
Name        string
```

...

}

Сохранение прогресса игрока

Rpc.meta:

```
RPC 51 SAVE_WORLD(  
    player_id : string  
    data      : DataWorld  
)  
    error : EnumRpcError  
end
```



rpc.go:

```
func (self *Rpc) RPC_SAVE_WORLD(p *RPC_SAVE_WORLD) {  
    ...  
    p.Out.Error = SaveData(p.in.PlayerId, &p.in.Data)  
}
```


Особенности языка

- ВМЕСТО КЛАССОВ **ТИПЫ**

```
type Reader interface {  
    Read() (error, []byte)  
}
```

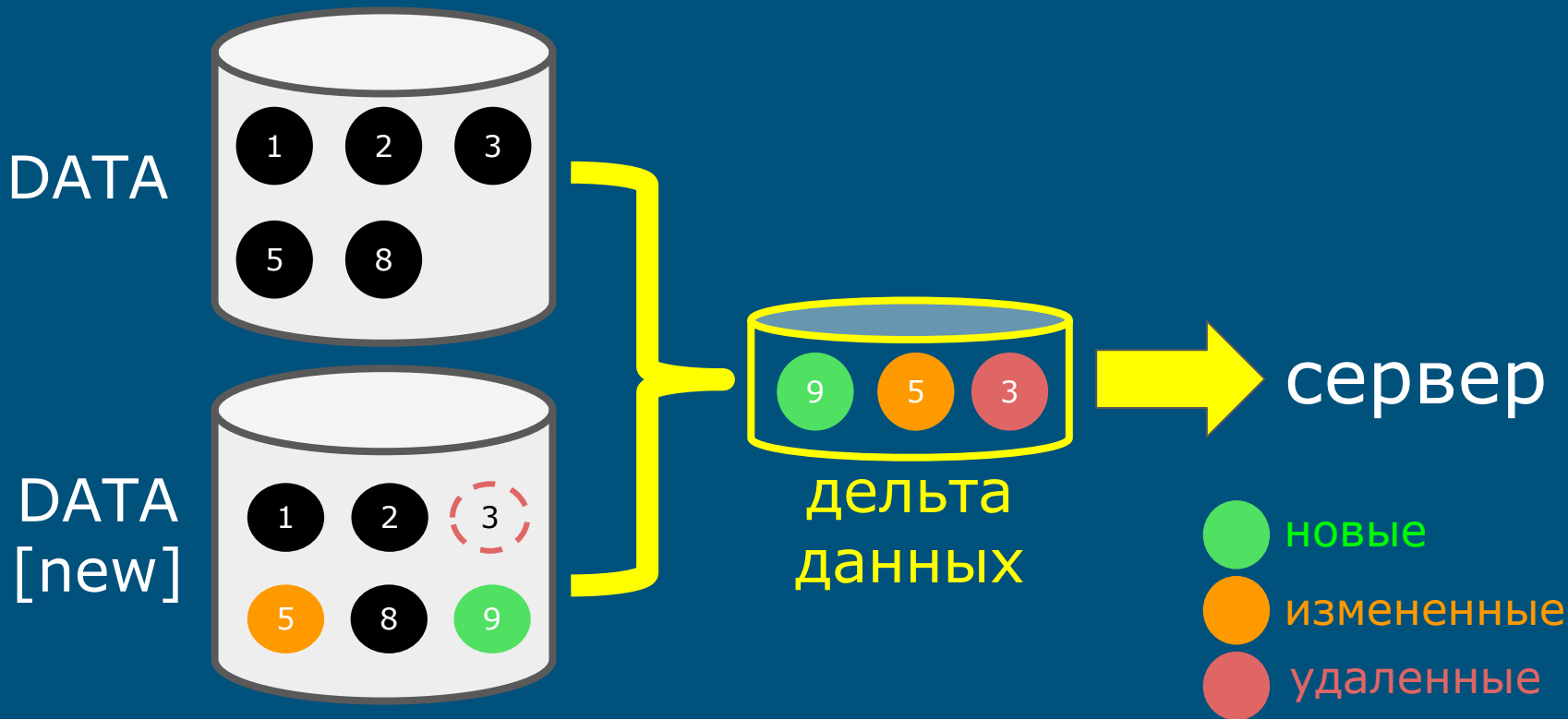
```
type Writer interface {  
    Write([]byte) error  
}
```

```
type RW struct {  
    Reader  
    Writer  
}
```

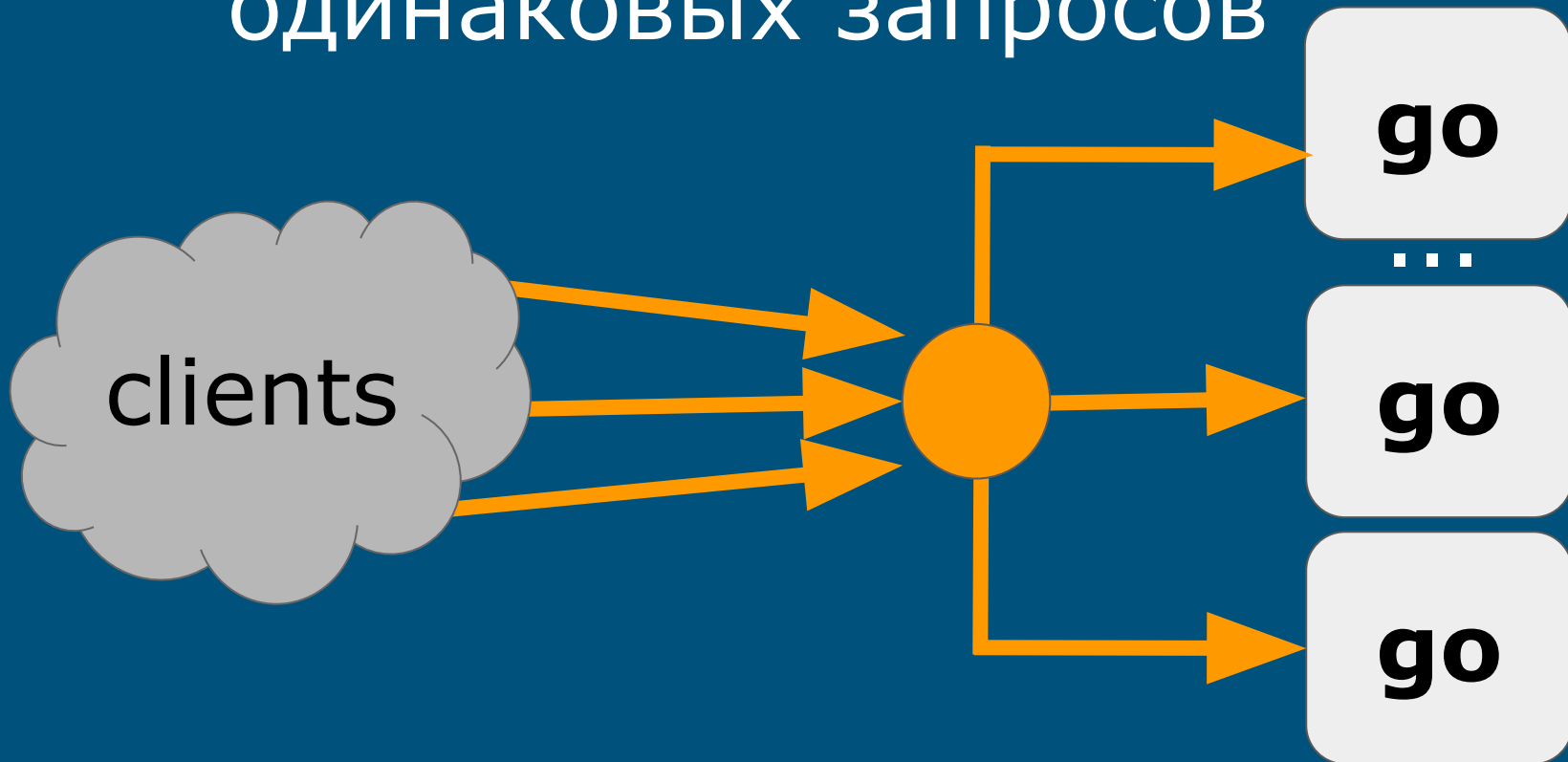
```
func (rw *RW) Read() (error, []byte) {  
    ...  
}
```

```
func (rw *RW) Write([]byte) error {  
    ...  
}
```

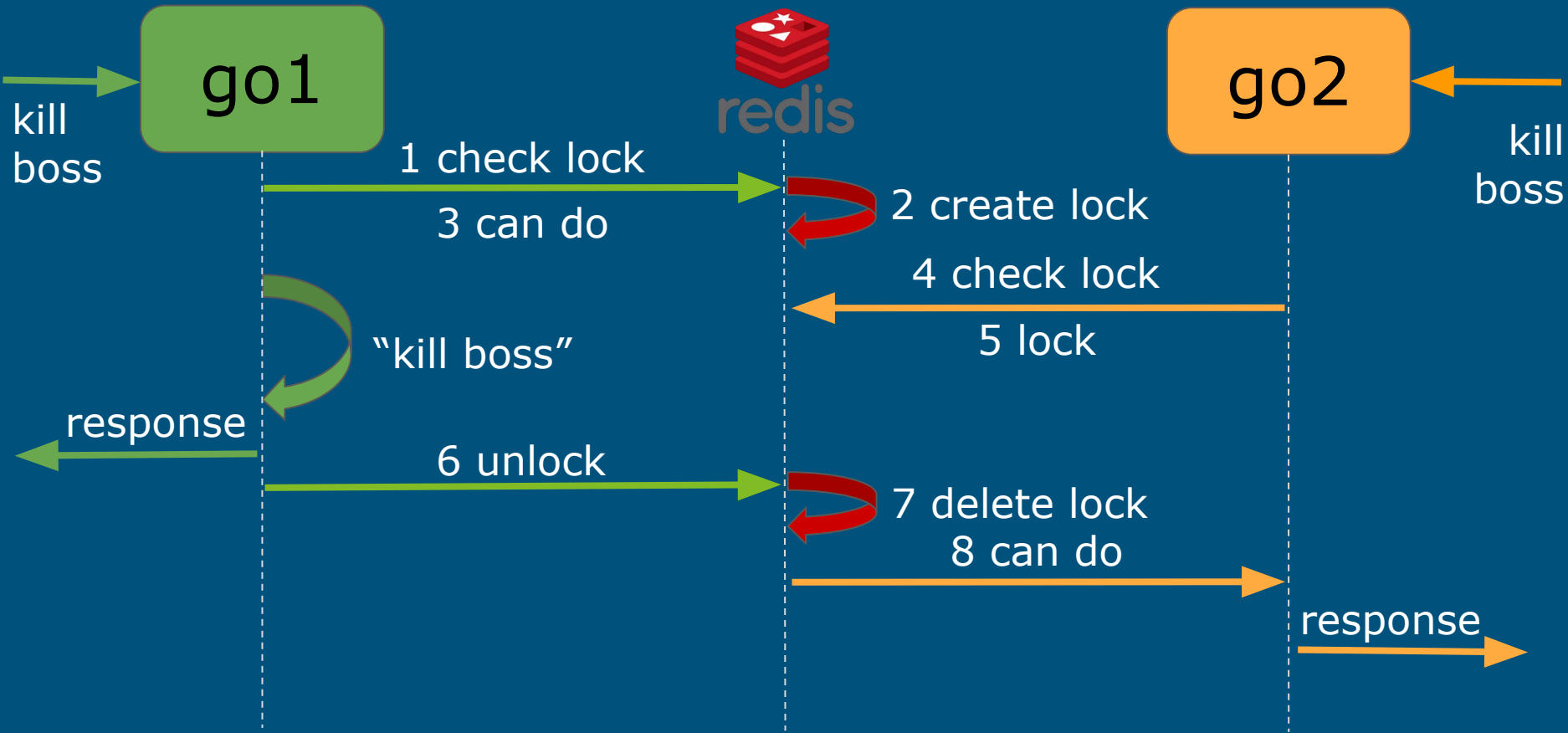
Сохранение изменений



Ошибка с повторной обработкой одинаковых запросов



Distributed lock



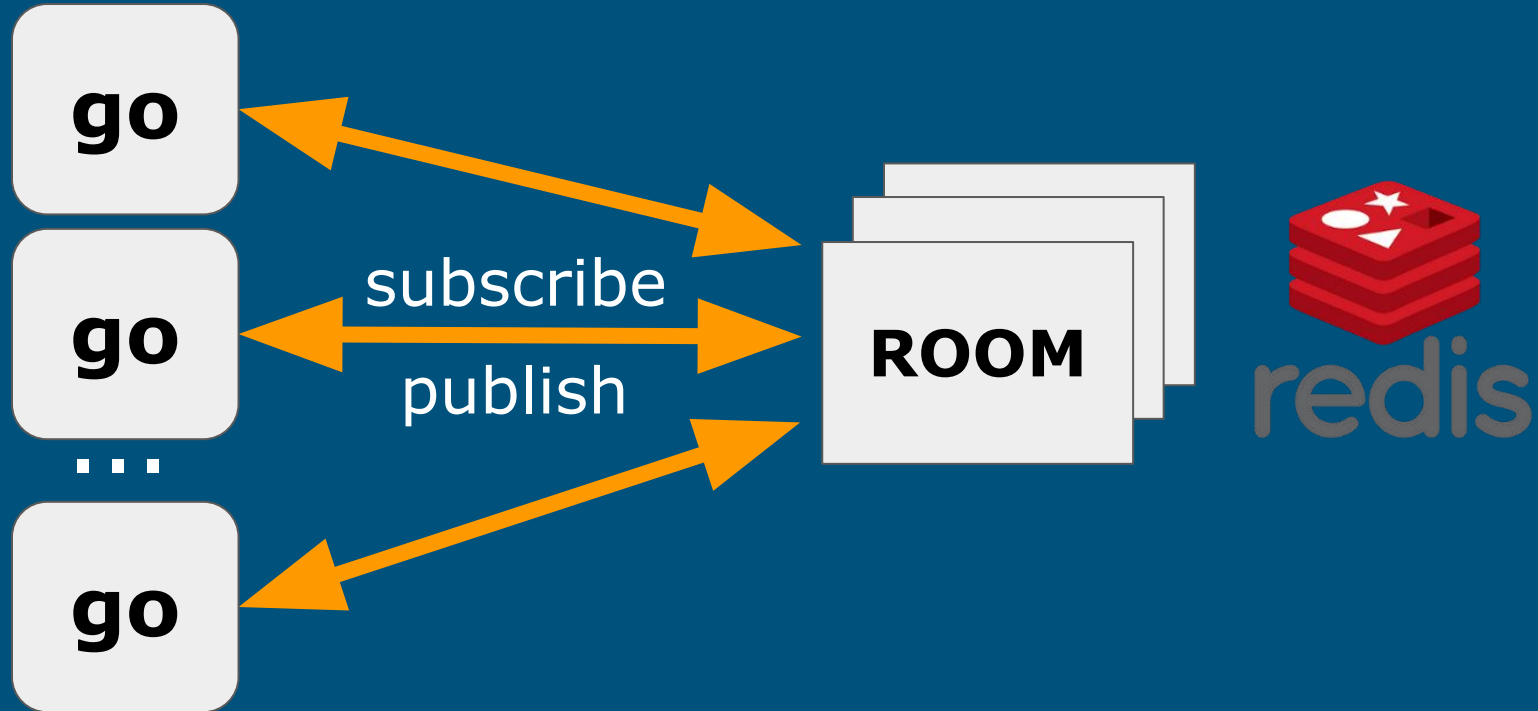
Distributed lock

github.com/hjr265/redsnc

```
func (self *Guild) EndBossFight(data *DataFight) error {  
    mutex := self.env.GetRdMutex("guild_end_fight", data.GuildId)  
    err := mutex.Lock()  
    if err != nil {  
        return err  
    }  
    defer mutex.Unlock()  
  
    // do end fight  
    ...  
}
```

Чат

- rooms: global, guild, private



Data race

```
var values []int

func main() {
    for i := 0; i < 3; i++ {
        go addValue(i)
    }
    time.Sleep(2 * time.Second)
    fmt.Println("values =", values)
}

func addValue(i int) {
    values = append(values, i)
}
```

```
$ go run race.go
values = [0 2 1]
```

```
$ go run -race race.go
=====
WARNING: DATA RACE
Read at 0x0000006191e0 by goroutine 8:
  main.addValue() example/race.go:99 +0x3d
Previous write at 0x0000006191e0 by goroutine 7:
  main.sum()      example/race.go:99 +0x59
```

Data race

```
var values []int
var mutex sync.RWMutex

func main() {
    for i := 0; i < 3; i++ {
        go addValue(i)
    }
    time.Sleep(2*time.Second)
    fmt.Println("values=", getValues())
}
```

```
func addValue(i int) {
    mutex.Lock()
    defer mutex.Unlock()
    values = append(values, i)
}
```

```
func getValues() []int {
    mutex.RLock()
    defer mutex.RUnlock()
    return values
}
```

```
$ go run -race race_mutex.go
values = [2 0 1]
```


Data race

```
func main() {  
    ch = make(chan int, 1)  
    for i := 0; i < 3; i++ {  
        go addValue(i)  
    }  
}
```

```
    fmt.Print("values = ")  
    for m := range ch {  
        fmt.Print(" ", m)  
    }  
}
```

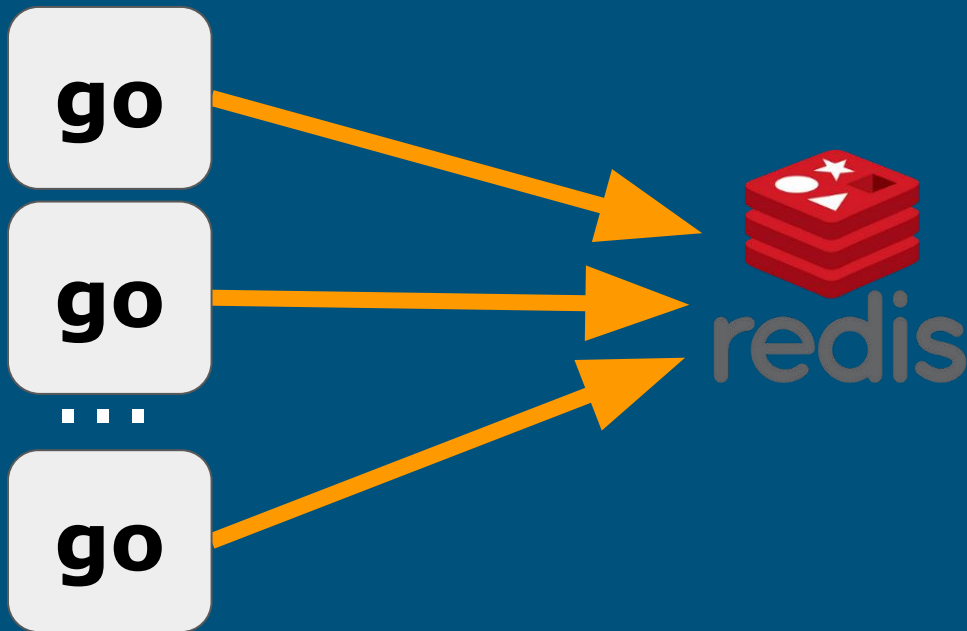
```
var ch chan int
```

```
func addValue(i int) {  
    ch <- i  
}
```

```
$ go run -race race_chan.go  
values = 2 0 1
```

Нотификации

- внутриигровые нотификации
- пуши



Антипаттерн с Wait group

```
func runWorker() {  
    for {  
        wg.Add(NotifQueueCapacity)  
        wg.Wait()  
        go sendNotifications()  
    }  
}
```



wg.Done()

```
func addNotification(item Notify) {  
    ...  
    wg.Done()  
}
```

```
gme/notification.go:97  
panic: sync: negative  
WaitGroup counter
```

КОМПИЛЯЦИЯ

- go install

```
$ go install -race -v bit.games/autogen gme/data  
gme/db gme/billing github.com/go-sql-driver/mysql
```

```
...
```

```
gamed/src/gme/guild.go:1223: no new variables on left side of :=  
gamed/src/gme/guild.go:1223: raid.Boom undefined (type *autogen.DataRaid has no  
field or method Boom)  
gamed/src/gme/guild.go:1223: multiple-value self.loadBossFight() in single-value context  
gamed/src/gme/guild.go:1227: undefined: fight  
gamed/src/gme/guild.go:1234: cannot assign to fight.Damge  
gamed/src/gme/guild.go:1235: undefined: fight in fight.Fights_count  
gamed/src/gme/guild.go:1238: too many errors
```

Кросс-компиляция

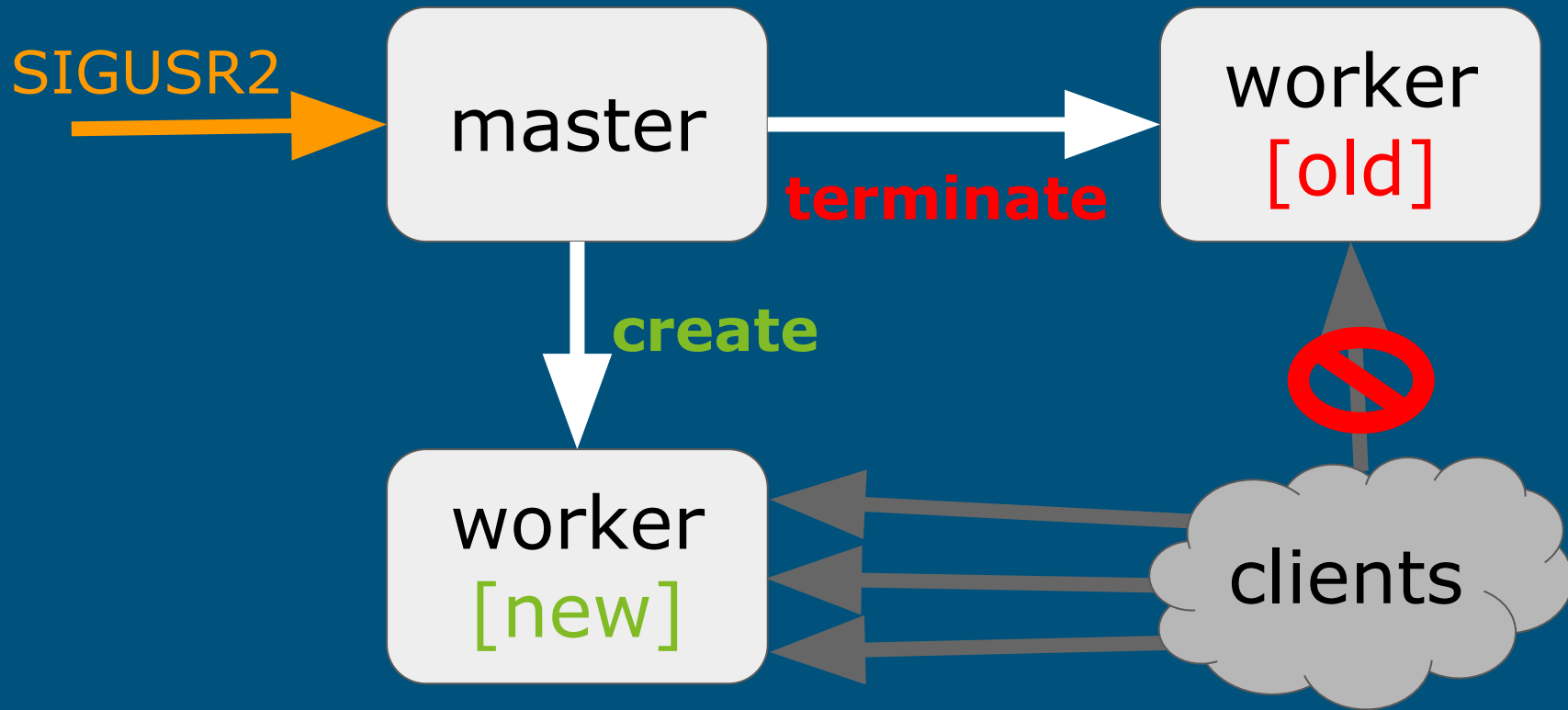
github.com/mitchellh/gox

```
$ gox -rebuild -output=./app -osarch="darwin/amd64" ...  
$ gox -rebuild -output=./app -osarch="linux/amd64" ...
```

```
package pcre  
/*  
#cgo LDFLAGS: -lpcr  
#cgo CFLAGS: -I/opt/local/include  
#include <pcr.h>  
#include <string.h>  
*/  
import "C"
```

```
$ gox -rebuild -output=gamed -osarch="linux/amd64" ...  
"Stderr: go build github.com/glenn-brown/pcr: no buildable Go  
source files in github.com/glenn-brown/pcr"
```

Деплой



Профилирование

- `net/http/pprof`
- `go tool pprof`
- `gorinba`

Профилирование

- net/http/pprof

```
$ lynx http://localhost:1234/debug/pprof
```

```
/debug/pprof/  
profiles:
```

```
    0 block  
  1828 goroutine  
 22627 heap  
    0 mutex  
    21 threadcreate
```

```
full goroutine stack dump
```


Профилирование

- pprof -> goroutines

```
goroutine profile: total 1828
```

```
479
# net.runtime_pollWait+0x58 runtime/netpoll.go:164
# net.(*pollDesc).wait+0x37 net/fd_poll_runtime.go:75
# net.(*pollDesc).waitRead+0x33 net/fd_poll_runtime.go:80
# net.(*netFD).Read+0x1b6 net/fd_unix.go:250
# net.(*conn).Read+0x6f net/net.go:181
# bufio.(*Reader).Read+0x311 bufio/bufio.go:213
# github.com/gorilla/websocket.(*Conn).readFull+0x90 github.com/gorilla/websocket/conn.go:546
# github.com/gorilla/websocket.(*Conn).advanceFrame+0x8e github.com/gorilla/websocket/conn.go:570
# github.com/gorilla/websocket.(*Conn).NextReader+0x56 github.com/gorilla/websocket/conn.go:705
# github.com/gorilla/websocket.(*Conn).ReadMessage+0x2e github.com/gorilla/websocket/conn.go:768
# gme.(*chatConnection).readPump+0x150 gme/chat.go:157
# gme.(*ChatHandler).ServeHTTP+0x1d6 gme/chat.go:1133
# net/http.(*ServeMux).ServeHTTP+0x12f net/http/server.go:2238
# bit.games/webserver.(*Server).ServeHTTP+0x9a bit.games/webserver/webserver.go:69
# net/http.serverHandler.ServeHTTP+0x91 net/http/server.go:2568
# net/http.(*conn).serve+0x611 net/http/server.go:1825
```

Профилирование

- go tool pprof

```
$ go tool pprof ./app http://127.0.0.1:5555/debug/pprof/profile
```

```
(pprof) top5
```

```
5055.75kB of 5763.69kB total (87.72%)
```

```
Dropped 487 nodes (cum <= 28.82kB)
```

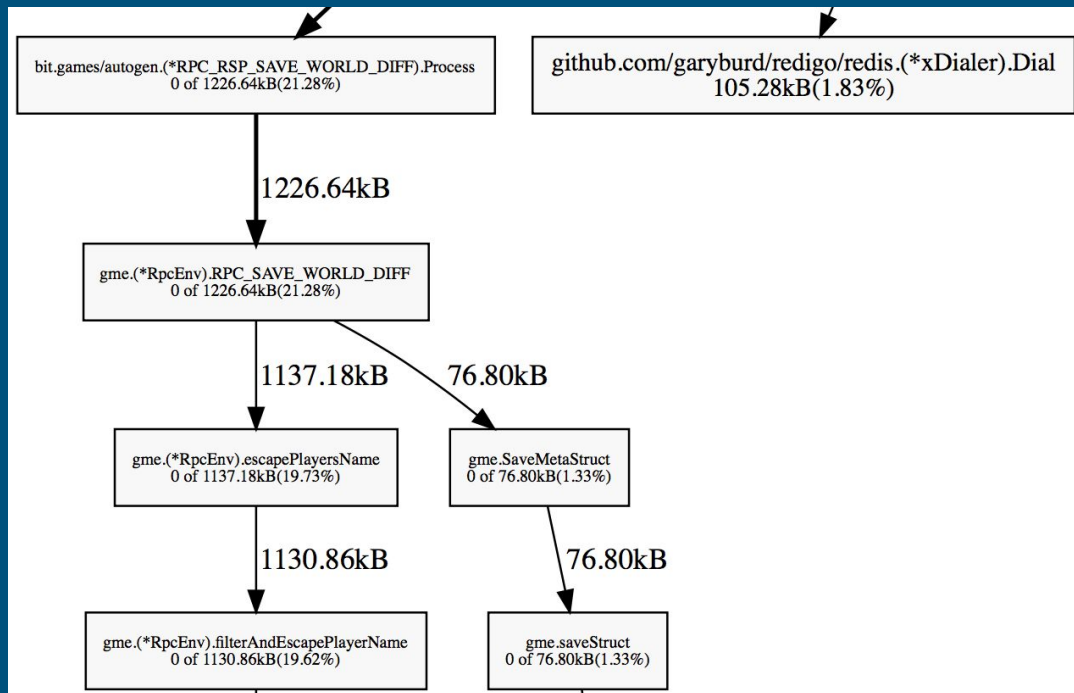
```
Showing top 5 nodes out of 62 (cum >= 287.56kB)
```

flat	flat%	sum%	cum	cum%	
2747.38kB	47.67%	47.67%	2747.38kB	47.67%	runtime.stringtoslicebyte
1122.73kB	19.48%	67.15%	1122.73kB	19.48%	github.com/dlclark/regexp2.(*runner).initMatch
551.22kB	9.56%	76.71%	3706.88kB	64.31%	gme.InitConfigs.func1
384kB	6.66%	83.37%	384kB	6.66%	runtime.hashGrow
250.42kB	4.34%	87.72%	287.56kB	4.99%	github.com/dlclark/regexp2/syntax.(*writer).codeFromTree

```
(pprof)
```


Профилірование

- go tool pprof



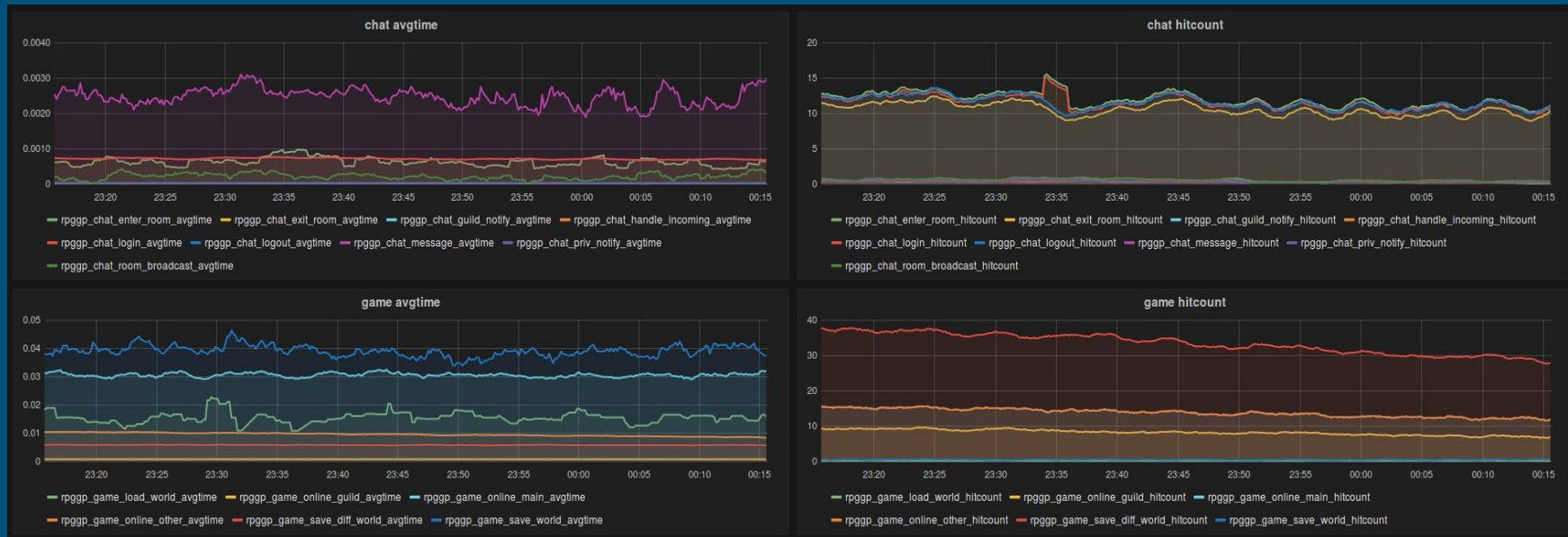
Профилирование

- gopinba (github.com/mkevac/gopinba)

```
func (self *Guild) EndBossFight(data *DataFight) error {  
    timer := self.env.PinbaStartTimer(TAG_GUILD, "end_fight")  
    defer timer.Stop()  
  
    // do boss fight  
    ...  
}
```

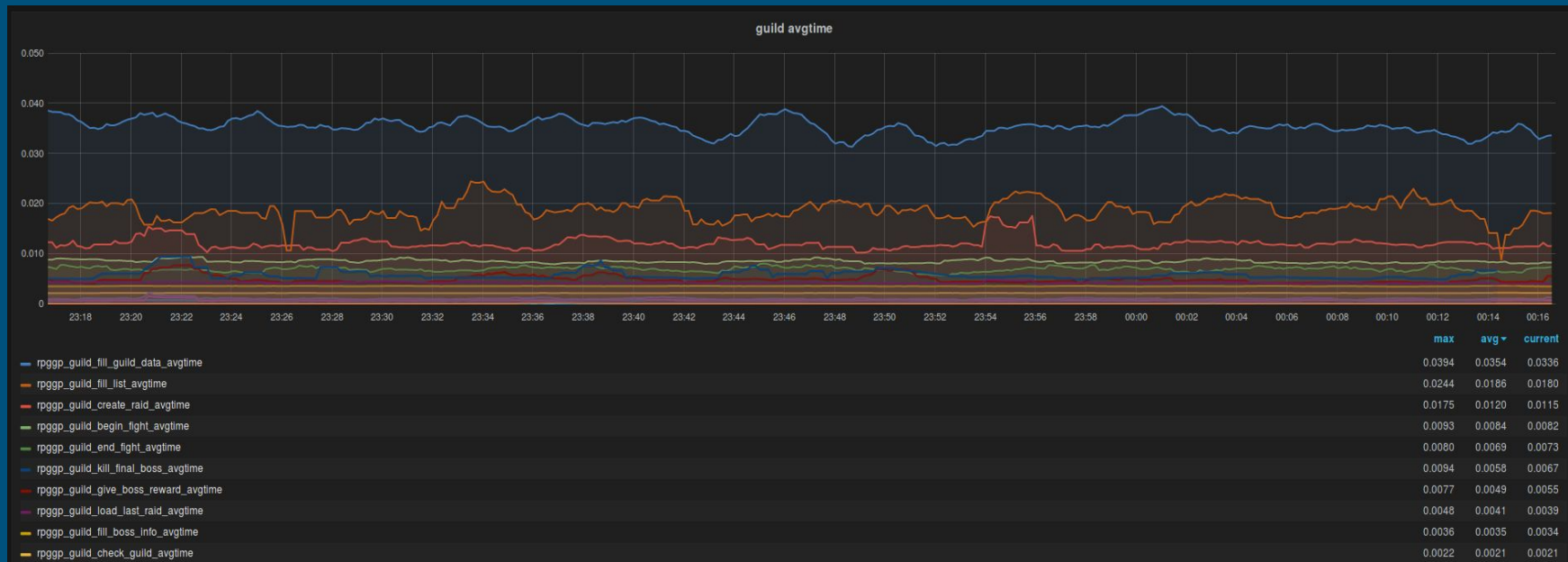
Профилирование

- gopinba (github.com/mkevac/gopinba)



Профилирование

- gorinba (github.com/mkevac/gorinba)



УТИЛИТЫ

- go get

```
$ go get github.com/go-sql-driver/mysql
$ go get github.com/mkevac/gopinba
$ go get golang.org/x/net
```

```
import (  
    "github.com/go-sql-driver/mysql"  
    "github.com/mkevac/gopinba"  
    "golang.org/x/net"  
)
```


УТИЛИТЫ

- gofmt

```
$ gofmt db/sharding.go
```

```
func (self *DB) ReleaseTransaction() error {  
    if self.Trx != nil  
    {  
if err := self.Trx.Rollback(); err != nil {  
    return err  
    }  
    }  
    self.Trx = nil  
    return nil  
}
```



```
func (self *DB) ReleaseTransaction() error {  
    if self.Trx != nil {  
        if err := self.Trx.Rollback(); err != nil {  
            return err  
        }  
    }  
    self.Trx = nil  
    return nil  
}
```

УТИЛИТЫ

- VIM (**fatih/vim-go**)

```
1247 err fight := self.loadBossFight(raid.Id)
1248 if err != nil {
1249     return err, EnumGuildResult_SYSTEM_ERR
1250 }
1251 if fight.Is_active != EnumRaidPlayerStatus_ACTIVE {
1252     return nil, EnumGuildResult_NO_FOUND_FIGHT
1253 }
1254
1255 fight.Damage += damage
1256 fight.Fights_count++
1257
1258 if damage >= raid.Health
1259     raid.Health = 0
1260 } else {
1261     raid.Health -= damage
```

/Volumes/SSD/var/rpg/default/server/gamed/src/gme/guild.go:1250,

```
1 guild.go|1247 col 6| : expected ';', found 'IDENT' fight
2 guild.go|1251 col 21| : expected '{', found '!'
3 guild.go|1252 col 3| : expected operand, found 'return'
4 guild.go|1259 col 15| : expected '==', found '='
```

Особенности языка

- статически типизированный язык
- высокая производительность
- го-рутины, каналы
- строгий синтаксис
- вместо классов используются типы, реализующие различные интерфейсы
- один бинарный файл для деплоя
- мощный инструментарий для профилирования
- большое количество утилит

Спасибо за внимание!
Вопросы?



Цаль-Цалко Иван
ivhome@bit-creative.com

