



#SECONRU



МЕЖРЕГИОНАЛЬНАЯ КОНФЕРЕНЦИЯ
РАЗРАБОТЧИКОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

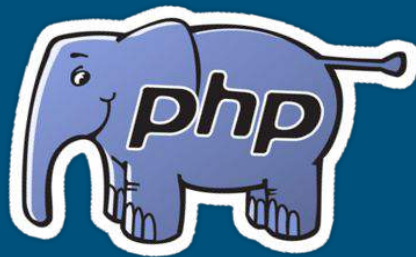
Go на практике



Цаль-Цалко Иван



Почему мы решили написать сервер на Go?



Особенности

- go прививает определенный **СТИЛЬ**

```
bin/
hello                # command executable
outyet               # command executable
pkg/
  linux_amd64/
    github.com/golang/example/
      stringutil.a    # package object
src/
  github.com/golang/example/
    .git/             # Git repository metadata
    hello/
      hello.go        # command source
    outyet/
      main.go         # command source
      main_test.go    # test source
    stringutil/
      reverse.go      # package source
      reverse_test.go # test source
  golang.org/x/image/
    .git/             # Git repository metadata
    bmp/
      reader.go       # package source
      writer.go       # package source
... (many more repositories and packages omitted) ...
```

```
// Method for printing - sorts the elements before printing.
func (s Sequence) String() string {
    sort.Sort(s)
    str := "["
    for i, elem := range s {
        if i > 0 {
            str += " "
        }
        str += fmt.Sprint(elem)
    }
    return str + "]"
}
```

Особенности

- передача **нескольких** значений в качестве результата

```
func (self ConfigManager) getById(id uint32) (error, meta.IMetaStruct) {  
    item, ok := self.id2conf[id]  
    if ok {  
        return nil, item  
    }  
    return errors.New(fmt.Sprintf("Not found config by id: %d", id)), nil  
}
```

Особенности

- функция **defer**

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

Особенности

- обработка ошибок на **верхнем** уровне

```
func (self *rpcHandler) doRequest(req Request, writer Writer) {
    resp, err := handleRPC(req)
    if err != nil {
        logger.Log(fmt.Sprintf("Error: '%s'", err))
        writer.WriteErr(err)
        return
    }
    ...
}

func handleRPC(req Request) ([]bytes, error) {
    ...
    players, err := loadPlayers(req)
    if err != nil {
        return resp, err
    }
    ...
}
```

Особенности

- обработка исключительных ситуаций (**panic**)

```
func (self *rpcHandler) doRequest(req Request, writer Writer) {
    defer func() {
        if err := recover(); err != nil {
            logger.Log(fmt.Sprintf("Error: '%s'", err))
            writer.WriteErr(err)
        }
    }()

    resp, err := handleRPC(req)
    ...
}

func handleRPC(req Request) ([]bytes, error) {
    ...
    players, err := loadPlayers(req)
    CheckErr(err)
    ...
}

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

Особенности

- статически типизированный язык
- ошибки только в продакшене
- нельзя быстро поменять что-то в продакшене :)

Utils

- go get
- gofmt
- golint
- go vet
- go test
- go install
- etc...

Utils

- 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"  
)
```

Utils

- gofmt

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

```
func (self *BitDB) StartTransaction() error {var err error
self.Trx, err=self.DbSession().Begin()
return err}

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

```
func (self *BitDB) StartTransaction() error {
    var err error
    self.Trx, err = self.DbSession().Begin()
    return err
}

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

Utils

- golint

```
$ golint ./secon.go
```

```
secon.go:739:1: exported function Filter should have comment or be unexported
```

```
secon.go:740:2: can probably use "var vsf []string" instead
```

```
secon.go:750:1: exported function Map should have comment or be unexported
```

```
secon.go:910:19: should drop = 0 from declaration of var ops; it is the zero value
```

```
secon.go:1177:6: don't use underscores in Go names; func worker_chan should be  
workerChan
```

```
secon.go:1353:3: should replace i += 1 with i++
```

```
secon.go:1521:8: should omit type string from declaration of var a; it will be inferred from  
the right-hand side
```

Utils

- go vet

```
$ go vet ./gamed/
```

```
gamed/gamed.go:423: the cancel function returned by context.WithCancel should be  
called, not discarded, to avoid a context leak
```

```
gamed/gamed.go:439: arg authorized for printf verb %d of wrong type: bool
```

```
exit status 1
```

Utils

- go test

```
$ go test .  
--- FAIL: TestFileSystemAvatar (0.00s)  
    avatar_test.go:59: FileSystemAvatar.GetAvatarURL should not return an error  
    avatar_test.go:63: FileSystemAvatar.GetAvatarURL wrongly returned  
FAIL  
FAIL    goblintprints/chat    0.011s
```

```
$ go test .  
ok     goblintprints/chat    0.012s
```

Utils

- go test -bench=Counter

```
$ go test -bench=Counter -cpu=4,16
```

BenchmarkNaiveCounter-4	300000000	5.19 ns/op
BenchmarkNaiveCounter-16	300000000	5.20 ns/op
BenchmarkMutexCounter-4	10000000	172 ns/op
BenchmarkMutexCounter-16	10000000	155 ns/op
BenchmarkChanCounter-4	5000000	446 ns/op
BenchmarkChanCounter-16	3000000	444 ns/op
BenchmarkAtomicCounter-4	100000000	17.6 ns/op
BenchmarkAtomicCounter-16	100000000	18.1 ns/op
BenchmarkAtomicFloatCounter-4	20000000	72.8 ns/op
BenchmarkAtomicFloatCounter-16	20000000	77.6 ns/op
BenchmarkRwMutexCounter-4	10000000	200 ns/op
BenchmarkRwMutexCounter-16	10000000	197 ns/op
PASS		
ok	example	23.381s

Utils

- VIM

```
1222
1223     err, fight := self.loadBossFight(raid.Id)
1224     if err != nil {
1225         return err, EnumGuildResult_SYSTEM_ERR
1226     }
1227     if fight.Is_active != EnumRaidPlayerStatus_ACTIVE {
1228         return nil EnumGuildResult_NO_FOUND_FIGHT
1229     }
1230
1231     fight.Damage + damage
1232     fight.Fights_count += 1
1233
1234     if damage >= raid.Health
1235         raid.Health = 0
1236     } else {
1237         raid.Health -= damage
1238     }
1239
1240     *boss_health = raid Health
1241     *data_fight = *fight
1242
```

/Volumes/SSD/var/rpg/default/server/gamed/src/gme/guild.go:1227,2 55% byte:105 lang:en

```
1 guild.go:1228 col 14: expected ';', found 'IDENT' EnumGuildResult_NO_FOUND_FIGHT
2 guild.go:1235 col 15: expected '=', found '='
3 guild.go:1240 col 22: expected ';', found 'IDENT' Health
4 guild.go:1244 col 6: expected operand, found '!='
5 guild.go:1245 col 3: expected '{', found 'return'
6 guild.go:1295 col 117: expected ';', found '{'
```


Utils

- 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
```

Data struct

```
struct DataWorld
```

```
  player      : DataPlayer @diff  
  items       : DataItem[] @default:[] @diff  
  regions     : DataRegion[] @default:[] @diff  
  dungeons    : DataDungeon[] @default:[] @diff  
  gates       : DataGate[] @default:[] @diff  
  boosters    : DataBooster[] @default:[] @diff
```

```
  ...
```

```
end
```

```
struct DataPlayer
```

```
  @table:player @pkey:id @owner:id  
  id            : uint32  
  level         : uint32  
  version       : string @strmax:16  
  reg_time      : uint32 @default:0  
  name          : string @default:""
```

```
  ...
```

```
end
```

Data struct

```
type IDataStruct interface {  
    ClassId() uint  
    ClassName() string  
    ClassProps() *map[string]string  
    ClassFileds() []string  
    ReadFields(reader Reader) error  
    WriteFields(writer Writer, field string) error  
}  
  
type DataWorld struct {  
    Player DataPlayer  
    Items []*DataItem  
    Regions []*DataRegion  
    ...  
}  
  
type DataPlayer struct {  
    Id uint32  
    Level uint32  
    Xp uint32  
    ...  
}
```

Rpc struct

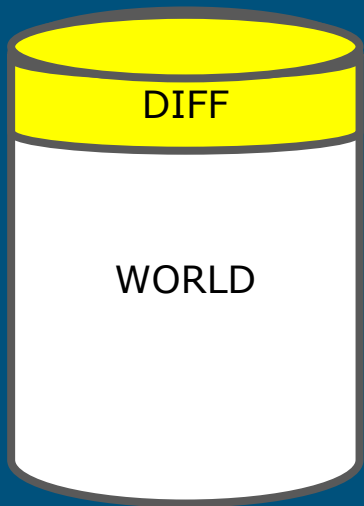
```
RPC 51 SAVE_WORLD(  
    plr_id : string @strmax:128  
    data   : DataWorld  
)  
    error   : EnumRpcError  
    version : uint32  
end
```

```
type IRpcPacket interface {  
    ClassId() uint  
    ClassName() string  
    ClassProps() *map[string]string  
    ClassFields() []string  
    ReadFields(reader Reader) error  
    WriteFields(writer Writer, field string) error  
    GetCode() int32  
    Process(interface{}, IMetaStruct) error  
}
```

Rpc.SaveWorld

```
func (self *Rpc) RpcSaveWorld(out *RpcRspSaveWorld, in *RpcReqSaveWorld) error {  
    ...  
    // check player by in.PlayerId  
    ...  
    return SaveMetaStruct(self.Db(), self.Logger(), &in.Data, in.PlayerId)  
}
```

Diff world



Data: 1, 2, 3, 5, 9

Data [new]: 1, 2, 5, 9, 10

Diff:

changed: 1, 5, 10

removed: 3

DataDiff struct

```
struct DataWorldDiff
  player      : DataPlayer[]
  items       : DataItem[]
  regions     : DataRegion[]
  dungeons    : DataDungeon[]
  gates       : DataGate[]
  ...
  removed_ids : DataRemovedIds[] @skip_save @skip_load
end
```

```
struct DataRemovedIds
  class_id : uint32
  ids      : uint32[]
end
```

Rpc.SaveWorldDiff

```
RPC 52 SAVE_WORLD_DIFF(  
    plr_id : string @strmax:128  
    data   : DataWorldDiff  
)  
    error   : EnumRpcError  
    version : uint32  
end
```

```
func (self *Rpc) RpcSaveWorldDiff(out *RpcRspSaveWorldDiff, in *RpcReqSaveWorldDiff) error {  
    ...  
    err := deleteByIds(self.Db(), self.Logger(), in.Player_id, in.RemovedIds)  
    ...  
  
    return SaveMetaStruct(self.Db(), self.Logger(), &in.Data, in.PlayerId)  
}
```


Data race

```
var value int

func main() {
    for i := 0; i < 3; i++ {
        go sum(i)
    }

    time.Sleep(3)
    fmt.Println("sum is ", value)
}

func sum(i int) {
    value = i + value
}
```

```
$ go run race.go
sum is 3
```

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

Data race

```
var value int
var mutex sync.RWMutex

func sum(i int) {
    mutex.Lock()
    defer mutex.Unlock()
    value += i
}

func getValue() int {
    mutex.RLock()
    defer mutex.RUnlock()
    return value
}

func main() {
    for i := 0; i < 3; i++ {
        go sum(i)
    }

    time.Sleep(2 * time.Second)
    fmt.Println("sum is ", getValue())
}
```

Data race

```
var ch chan int

func sum(i int) {
    ch <- i
}

func main() {
    ch = make(chan int)

    for i := 0; i < 3; i++ {
        go sum(i)
    }

    time.Sleep(2 * time.Second)

    sum := 0
    for i := 0; i < 3; i++ {
        sum += <- ch
    }
    fmt.Println("sum is ", sum)
}
```

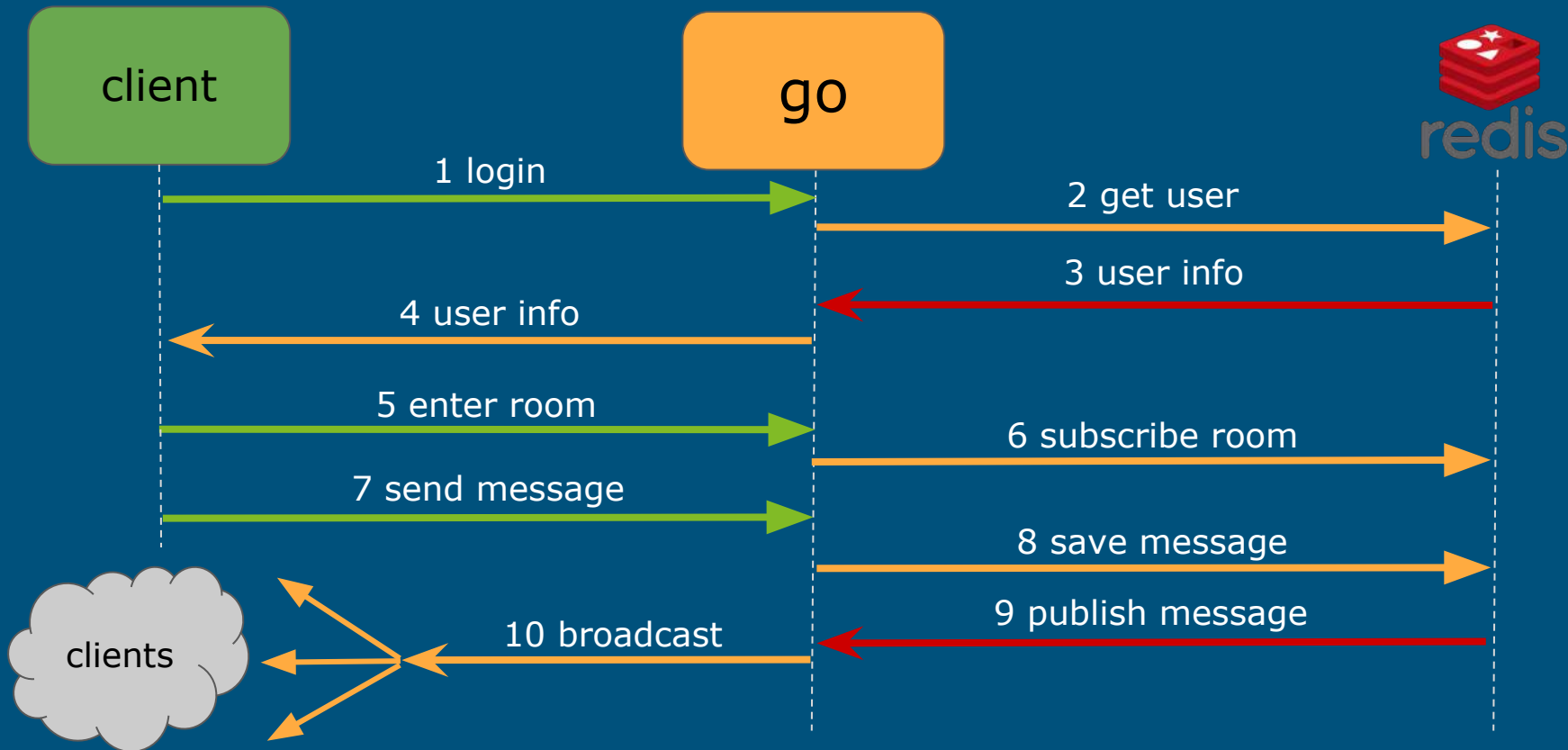
Chat

github.com/gorilla/websocket

github.com/garyburd/redigo/redis

- global rooms
- guild rooms
- private rooms

Chat



Chat

- handle incoming requests

```
func (h *chatHub) handleIncoming(cp *chatPacket) {
    if p, ok := cp.m.(*NetChatLogin); ok {
        h.handleLogin(cp.c, p)
    } else if p, ok := cp.m.(*NetChatRoomEnter); ok {
        h.handleEnterRoom(cp.c, p)
    } else if p, ok := cp.m.(*NetChatMsg); ok {
        h.handleMessage(cp.c, p)
    } else if p, ok := cp.m.(*NetChatCheckPrivates); ok {
        h.handleCheckPrivates(cp.c, p)
    } else if p, ok := cp.m.(*NetChatRoomExit); ok {
        h.handleExitRoom(cp.c, p)
    } else if _, ok := cp.m.(*NetChatLogout); ok {
        h.handleLogout(cp.c)
    } else {
        chlog(2, "unknown net message: %s", cp.m.CLASS_NAME())
    }
}
```

Chat

- subscribe to room

```
func (r *chatRoom) subscribeToRedis() chan bool {
    ready := make(chan bool, 1)
    chlog(0, "room '%s' subscribing to redis", r.key)
    if rdconn, err := getRedisFreshConn(); err == nil {
        r.psc = redis.PubSubConn{Conn: rdconn}
        r.psc.Subscribe("croompub:" + r.key)
        go func() {
            ready <- true
            close(ready)
            r.redisPump()
        }()
    } else {
        chlog(2, fmt.Sprintf("error: %v", err))
        ready <- false
        close(ready)
    }
    return ready
}
```

Chat

- broadcast message to all users in room

```
for {
    switch v := r.psc.Receive().(type) {
    case redis.Message:
        chlog(0, "got redis msg: len %v", len([]byte(v.Data)))
        go hub.handleBroadcastFromRedisRoom(r, []byte(v.Data))
    case redis.Subscription:
        // we don't need to listen to subscription messages,
    case error:
        chlog(0, "got redis error: %v", error(v))
        select {
        //in case we're done don't resubscribe
        case <-r.done:
            return
        //in case of error try to resubscribe to redis
        default:
            r.waitSubscribeToRedis()
        }
    }
    return
}
```


Cross compilation

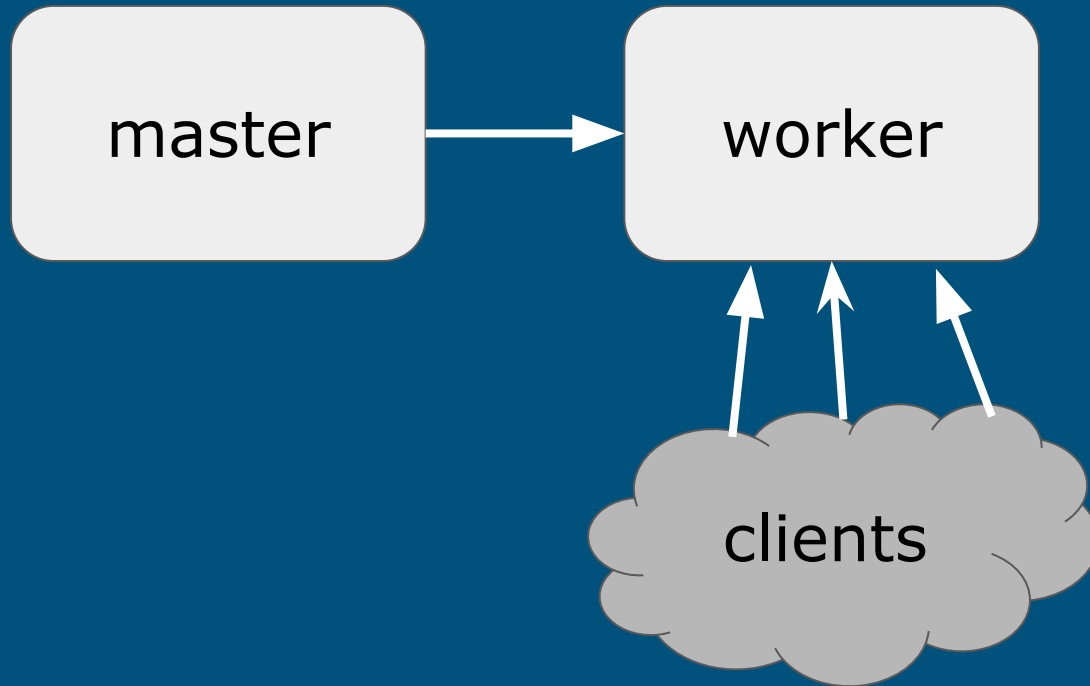
github.com/mitchellh/gox

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

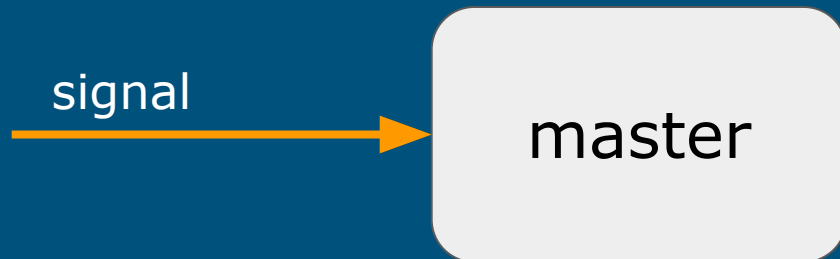
```
package pcre|  
/*  
#cgo LDFLAGS: -lpcre  
#cgo CFLAGS: -I/opt/local/include  
#include <pcre.h>  
#include <string.h>  
*/  
import "C"
```

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

Deploy server

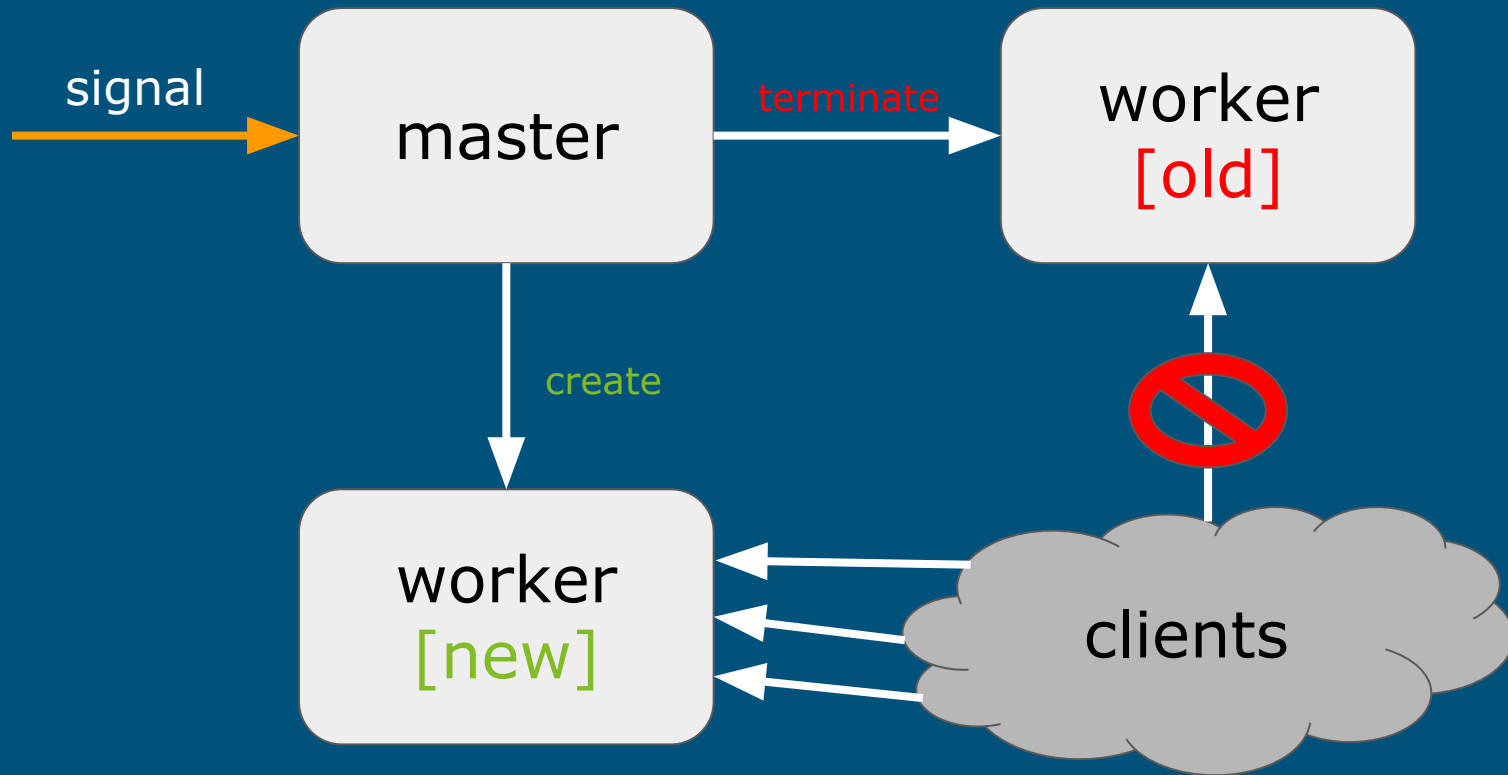


Deploy server

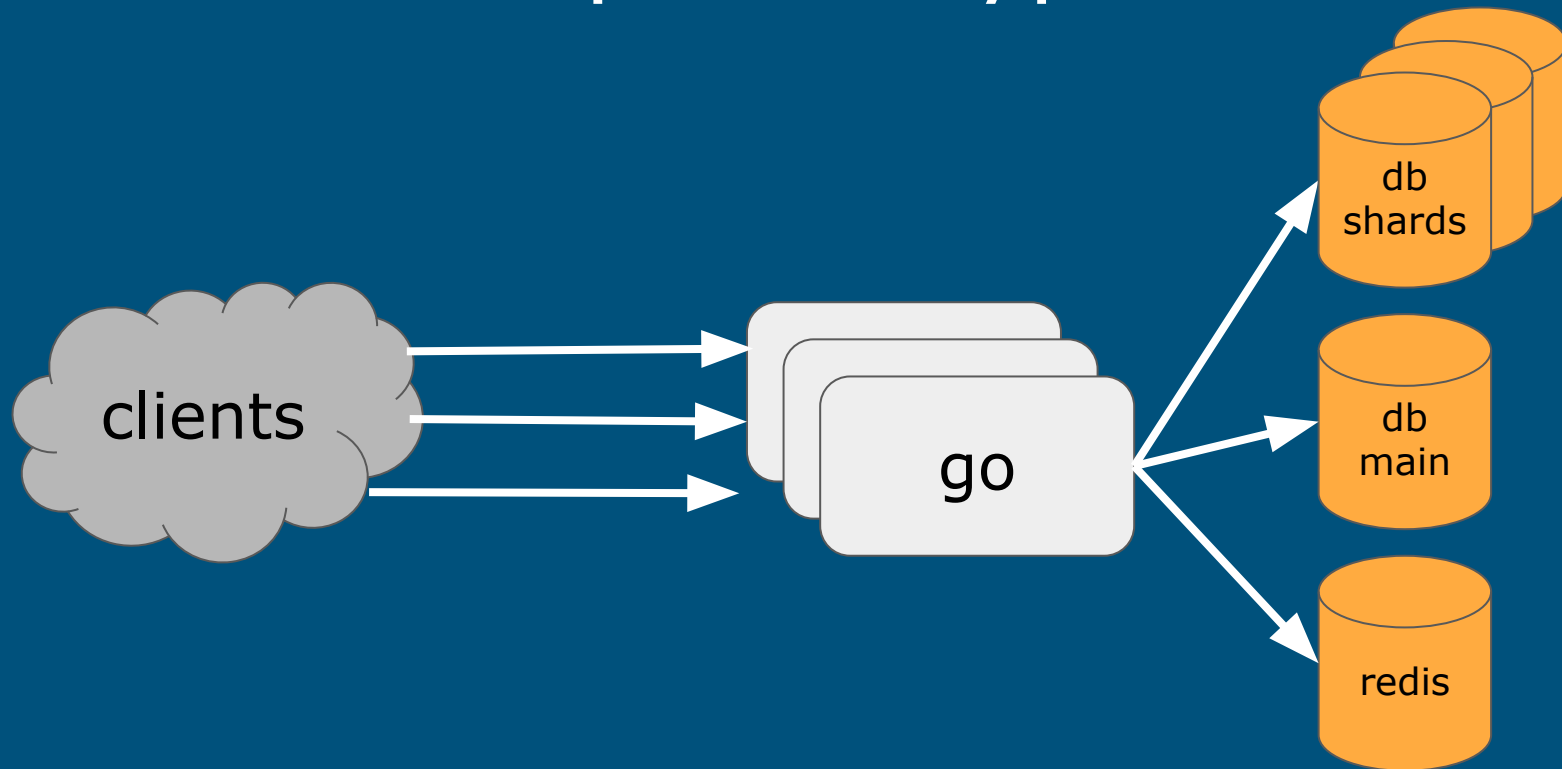


```
signal_chan = make(chan os.Signal, 10)
signal.Notify(signal_chan, syscall.SIGTERM, syscall.SIGUSR2)
log_main("Waiting for signals")
for {
    sig := <-signal_chan
    switch sig {
    case syscall.SIGTERM:
        log_main("Request for termination")
        terminateMaster()
        return
    case syscall.SIGUSR2:
        log_main("Request for binary reloading")
        reload_worker_chan <- true
    }
}
```

Deploy server



Архитектура



Distributed lock

github.com/hjr265/redsnc

```
func (self *Guild) EndBossFight(data *DataFight) (error) {
    timer := self.env.PinbaStartTimer(PINBA_TAG_GUILD, "end_fight")
    defer timer.Stop()

    mutex := self.env.GetRdMutex(fmt.Sprintf("guild_end_fight_%d", data.GuildId))
    err := mutex.Lock()
    if err != nil {
        return err
    }
    defer mutex.Unlock()

    //... do end fight
}
```

Profiling

- net pprof
- мониторинг памяти и GC
- github.com/divan/expvarmon
- go tool pprof
- gopinba

Profiling

- net pprof

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

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

```
0 block  
906 goroutine  
524 heap  
0 mutex  
14 threadcreate
```

```
full goroutine stack dump
```


Profiling

- pprof -> goroutines

```
goroutine profile: total 1044
```

```
479
# net.runtime_pollWait+0x58
# net.(*pollDesc).wait+0x37
# net.(*pollDesc).waitRead+0x33
# net.(*netFD).Read+0x1b6
# net.(*conn).Read+0x6f
# bufio.(*Reader).Read+0x311
# github.com/gorilla/websocket.(*Conn).readFull+0x90
# github.com/gorilla/websocket.(*Conn).advanceFrame+0x8e
# github.com/gorilla/websocket.(*Conn).NextReader+0x56
# github.com/gorilla/websocket.(*Conn).ReadMessage+0x2e
# gme.(*chatConnection).readPump+0x150
# gme.(*ChatHandler).ServeHTTP+0x1d6
# net/http.(*ServeMux).ServeHTTP+0x12f
# bit.games/webserver.(*Server).ServeHTTP+0x9a
# net/http.serverHandler.ServeHTTP+0x91
# net/http.(*conn).serve+0x611

runtime/netpoll.go:164
net/fd_poll_runtime.go:75
net/fd_poll_runtime.go:80
net/fd_unix.go:250
net/net.go:181
bufio/bufio.go:213
github.com/gorilla/websocket/conn.go:546
github.com/gorilla/websocket/conn.go:570
github.com/gorilla/websocket/conn.go:705
github.com/gorilla/websocket/conn.go:768
gme/chat.go:157
gme/chat.go:1133
net/http/server.go:2238
bit.games/webserver/webserver.go:69
net/http/server.go:2568
net/http/server.go:1825
```

Profiling

- pprof -> goroutines

...

32

# 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).fill+0x116	bufio/bufio.go:97
# bufio.(*Reader).ReadSlice+0xba	bufio/bufio.go:338
# github.com/garyburd/redigo/redis.(*conn).readLine+0x48	github.com/garyburd/redigo/redis/conn.go:228
# github.com/garyburd/redigo/redis.(*conn).readReply+0x42	github.com/garyburd/redigo/redis/conn.go:301
# github.com/garyburd/redigo/redis.(*conn).Receive+0x3b	github.com/garyburd/redigo/redis/conn.go:384
# github.com/garyburd/redigo/redis.PubSubConn.Receive+0x4b	github.com/garyburd/redigo/redis/pubsub.go:97
# gme.(*chatRoom).redisPump+0x101	gme/chat.go:240
# gme.(*chatRoom).subscribeToRedis.func1+0x61	gme/chat.go:318

Profiling

- мониторинг памяти и GC

```
type MemStats struct {  
    Alloc uint64  
    TotalAlloc uint64  
    Sys uint64  
    Lookups uint64  
    Mallocs uint64  
    Frees uint64  
    HeapAlloc uint64  
    HeapSys uint64  
    HeapReleased uint64  
    GCSys uint64  
    NextGC uint64  
    LastGC uint64  
    PauseTotalNs uint64  
    PauseNs [256]uint64  
    PauseEnd [256]uint64  
    NumGC uint32  
    NumForcedGC uint32  
    ...  
}
```

Profiling

- github.com/divan/expvarmon



Profiling

- 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)
```

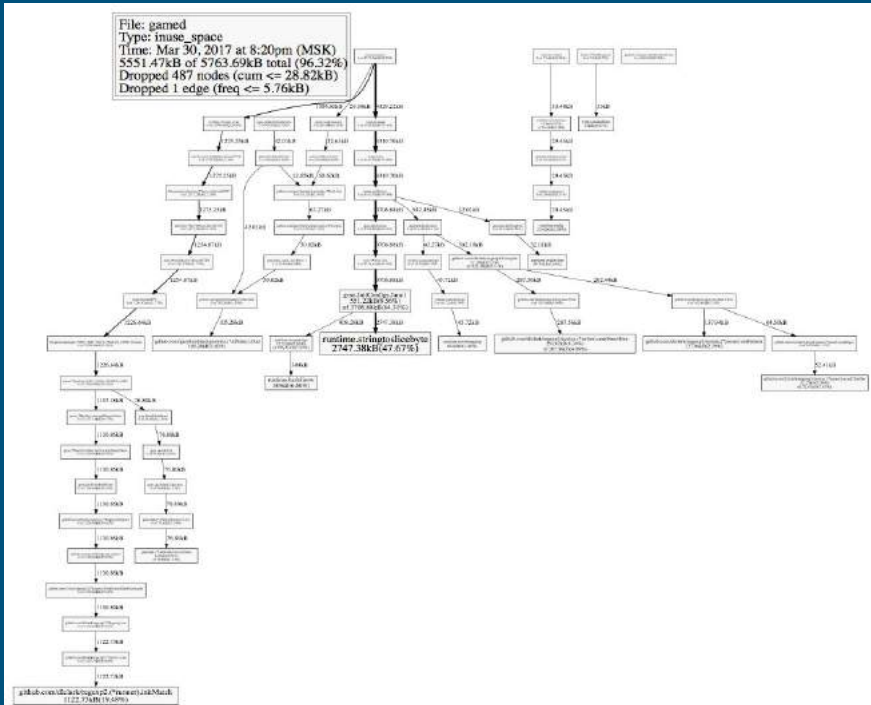
```
Showing top 5 nodes out of 62 (cum >= 287.56kB)
```

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)
```

Profiling

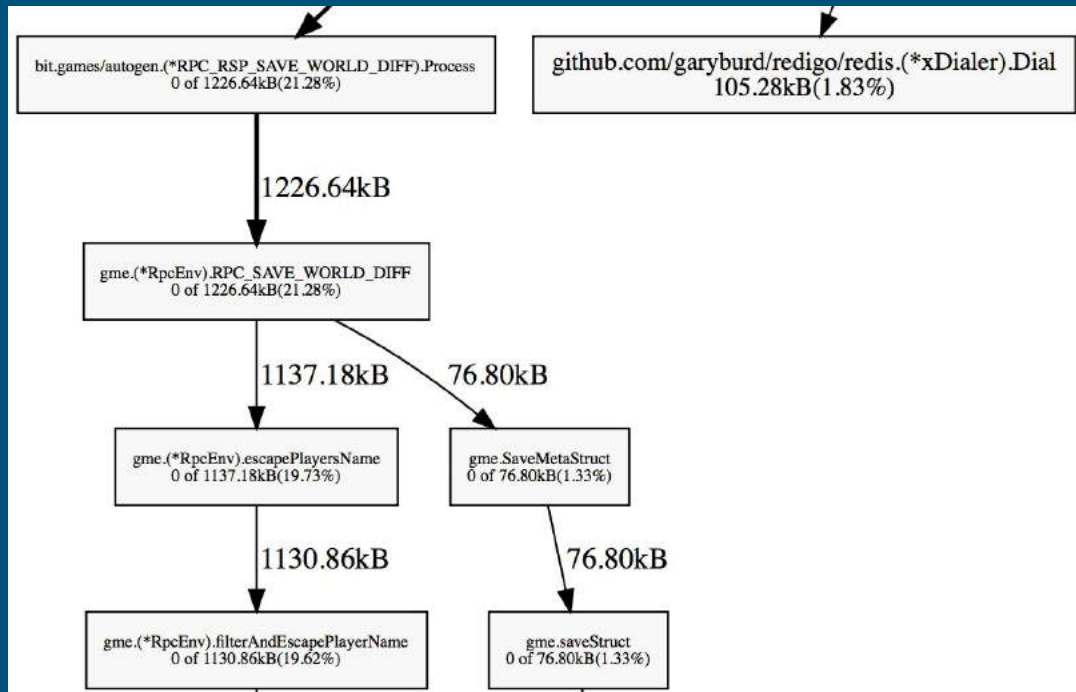
- go tool pprof



```
$ (pprof) web
```

Profiling

- go tool pprof



Profiling

- gopinba

github.com/mkevac/gopinba

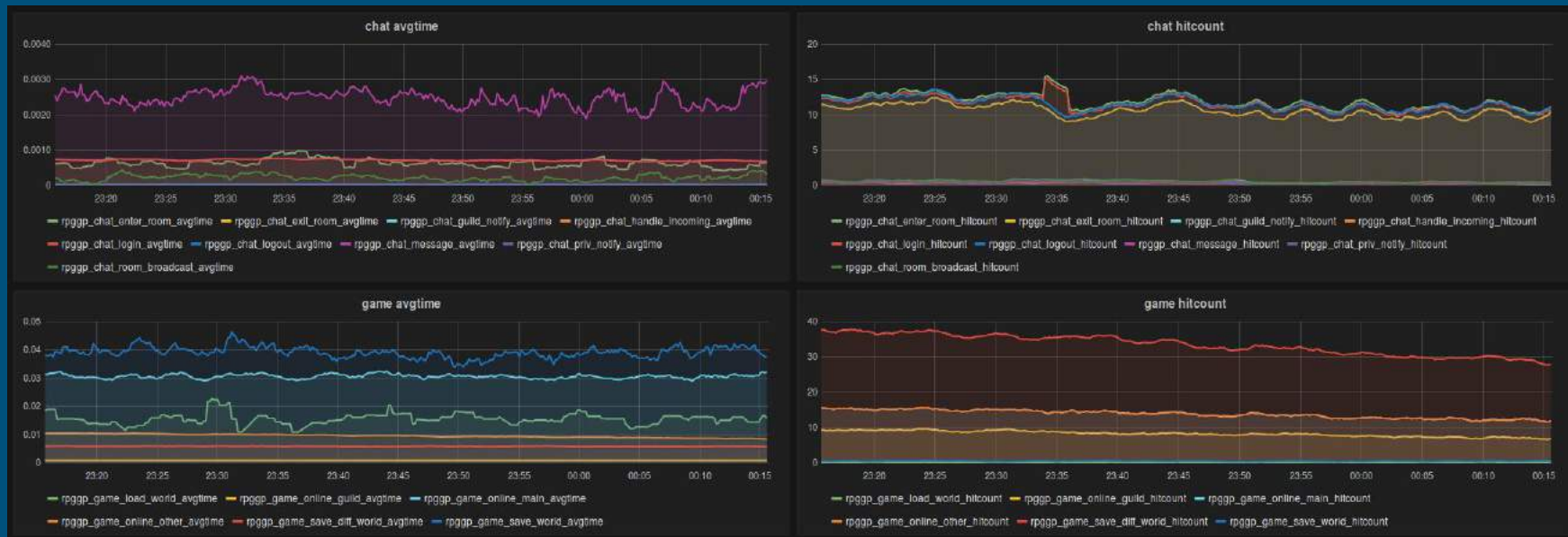
```
func (self *Guild) EndBossFight(data *DataFight) (error) {
    timer := self.env.PinbaStartTimer(PINBA_TAG_GUILD, "end_fight")
    defer timer.Stop()

    mutex := self.env.GetRdMutex(fmt.Sprintf("guild_end_fight_%d", data.GuildId))
    err := mutex.Lock()
    if err != nil {
        return err
    }
    defer mutex.Unlock()

    //... do end fight
}
```

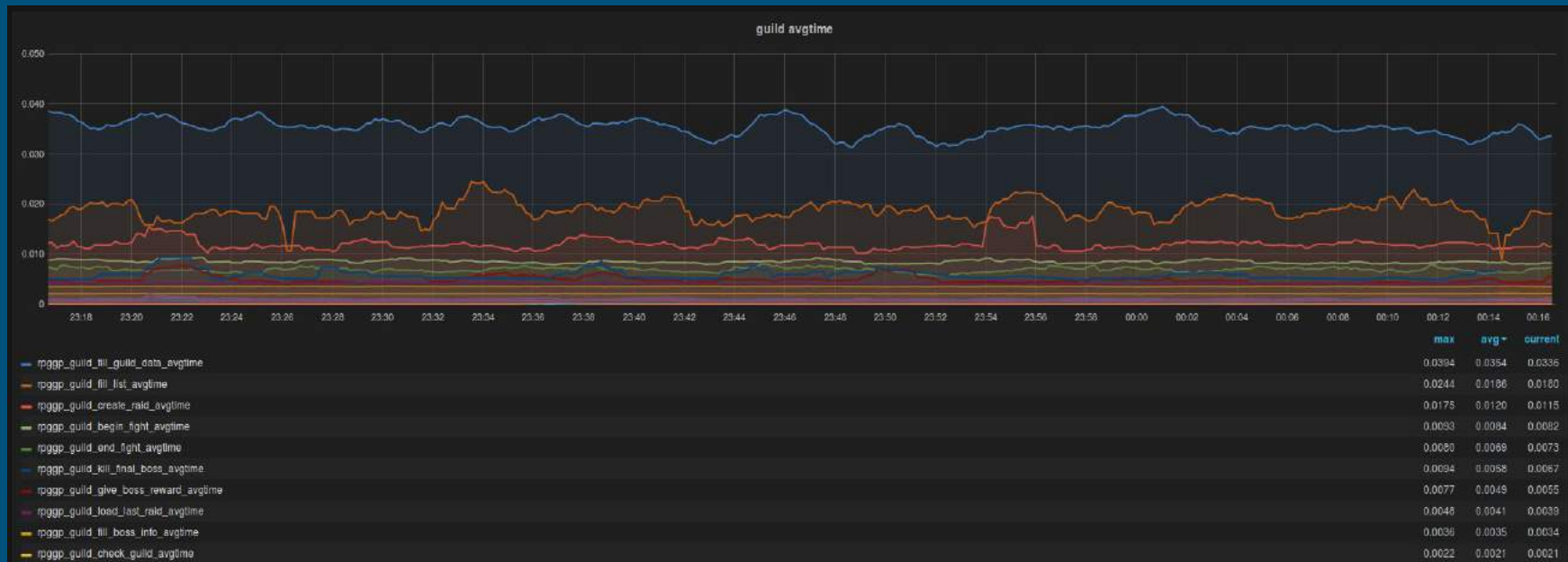

Profiling

- gopinba



Profiling

- gopinba



Ошибки

- data race
- distribute mutex
- отправка данных в закрытый канал
- переиспользование переменных в условных операторах
- `SelectBySQL(...).Where(...)`
- переиспользование Wait group

Ошибки

- переиспользование переменных в условных операторах

```
var sum int = 0
var value int = 0

if true {
    value := 50
    fmt.Println("value =", value)
}

sum += value

fmt.Println("value =", value)
fmt.Println("sum =", sum)
```

Output:

```
value = 50
value = 0
sum = 0
```

Ошибки

- SelectBySQL(...).Where(...)

```
var ids []uint32
err = db().SelectBySql("SELECT id FROM raid").Where("guild_id = ?", guild.Id).LoadValues(&ids)
if (err != nil) {
    return err
}
db().DeleteFrom("raid_players").Where("raid_id IN ?", ids).Exec()
```

Ошибки

- переиспользование Wait group

```
go func() {  
    for {  
        self.wait_group.Wait()  
        self.wait_group.Add(NOTIFICATION_QUEUE_CAPACITY)  
        go self.storeNotification()  
    }  
}()  
  
for {  
    switch dtype := self.rdconn.Receive().(type) {  
    case redis.Message:  
        self.wait_group.Done()  
    case redis.Subscription:  
        self.log(fmt.Sprintf("subscribe to rd: %v", dtype))  
    case error:  
        // handle error  
    }  
}
```

gme/notification.go:97 +0xc5

panic: sync: WaitGroup is reused before previous Wait has returned

Спасибо за внимание!
Вопросы?



Цаль-Цалко Иван
ivhome@bit-creative.com

