

ozon{tech

ozon{ech ozon{ech ozon
{ech ozon{ech ozon{ech
ozon{ech ozon{ech ozon
{ech ozon{ech ozon{ech
ozon{ech ozon{ech ozon



Профилирование и метрики в Go

Кирилл Шматов

старший разработчик платформы коммуникаций

Москва 2021



Профилирование

Добавление профилировщика к программе

```
include (  
    "net/http"  
    _ "net/http/pprof"  
)  
  
func main() {  
    go http.ListenAndServe("localhost:9000", nil)  
}
```

Просмотр текущего состояния

1. В браузере открыть <http://localhost:9000/debug/pprof>
2. `go tool pprof -http localhost:8081 http://localhost:9000`
3. `go tool pprof -http=":9090" -seconds=30 http://localhost:9000/debug/pprof/profile`

Получение из бенчмарков

```
$ GOGC=off go test -bench=functionName -cpuprofile cpu.out -memprofile mem.out  
$ go tool pprof cpu.out  
$ go tool pprof mem.out
```



Метрики

Подключение

```
include (  
    _ "expvars"  
    "net/http"  
)  
  
func main(){  
    go http.ListenAndServe("localhost:9000", nil)  
}
```

Просмотр

Открыть в браузере <http://localhost:9000/debug/vars>

По умолчанию возвращаются командная строка и текущие метрики памяти

Расширение метрик

```
package myvars

import (
    _ "expvars"
    "runtime"
)

// Возвращает количество горутин минус main
func numGoroutines() interface{} {
    return runtime.NumGoroutine() - 1
}

func init(){
    expvar.Publish("num_goroutines", expvar.Func(numGoroutines))
}
```



Почитать

1. [Профилирование и оптимизация веб-приложений на Go](#)
2. [The Complete Guide to Profiling Your Code](#)
3. [Профилирование и оптимизация программ на Go](#)
4. [pprof documentation](#)
5. [Expvarmon](#)
6. [ExpvarMon — консольный мониторинг сервисов на Go](#)
7. [Google](#)