

PERFORMANCE OPTIMISATION

HOW DO I GO ABOUT IT?

KAT ZIEŃ | @KASIAZIEN

GOTHAM GO

5-6 OCTOBER 2017



WHY

- fast is better than slow (duh)
- memory efficient is good
- saving money is good
- running out of memory in production is bad
- *suddenly* running out of memory in production is BAD

WHEN

- development
- load testing
- monitoring production

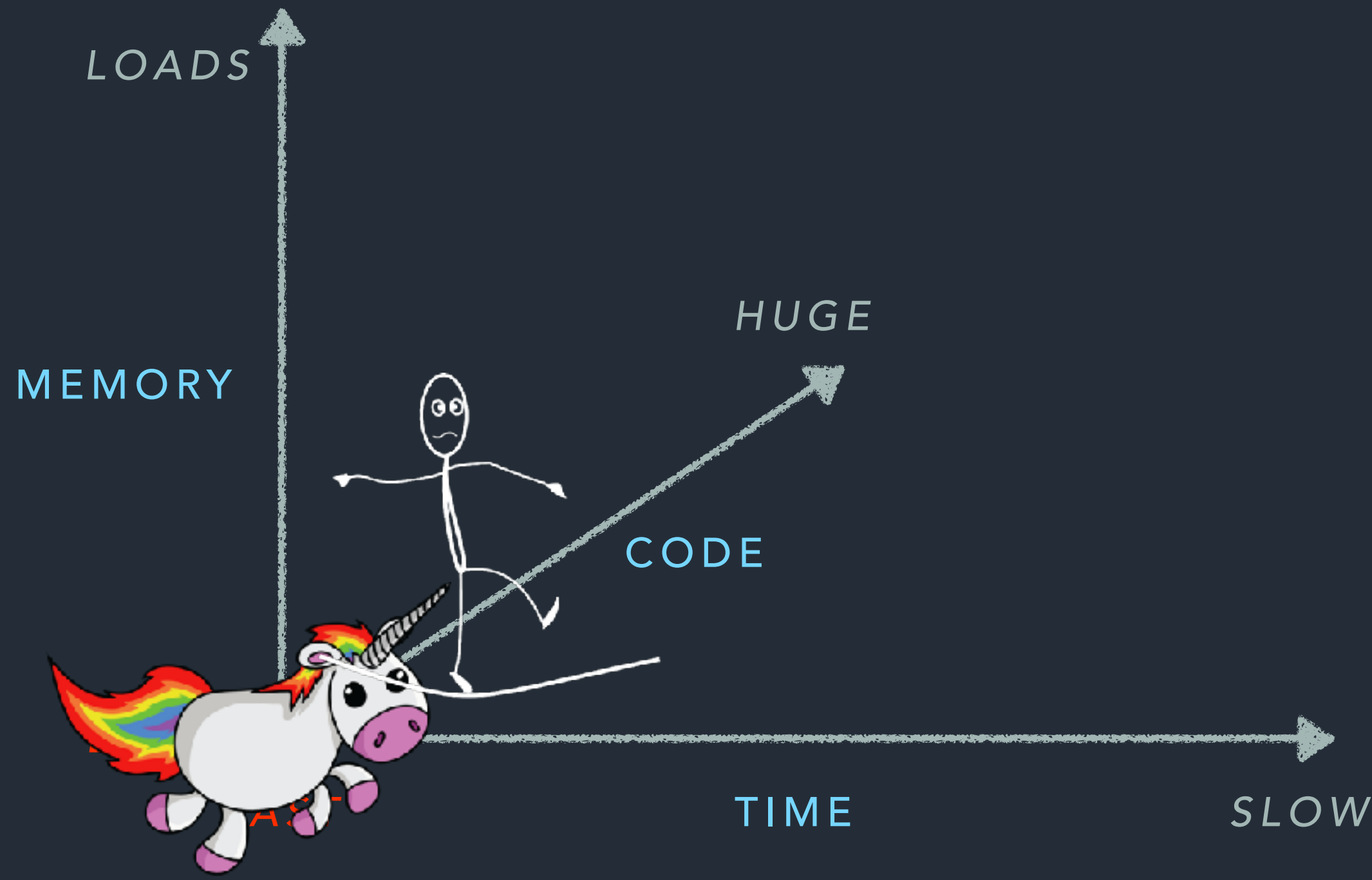
WHAT

“Do I need (to do) this?”



- execution time / no of operations
- memory usage / allocation

GOAL



- ▶ memory leaks
- ▶ race conditions
- ▶ deadlocks / livelocks
- ▶ heap allocations
- ▶ redundancy / inefficiency
- ▶ dead code / unused libraries

- Prefer stack over heap allocations
- Prefer values over pointers for not big structures
- Preallocate maps and slices if you know the size beforehand
- Don't log if you don't have to
- Use buffered I/O if you do many sequential reads or writes
- If your application extensively uses JSON, consider using parser/serialiser generators (e.g. easyjson)
- Every operation matters in a hot path

HOW

HOW???

- How do I know what to improve?
- How do I know what my app is doing?
- How do I know what my app can handle?

- How do I measure?
- How do I know what to measure?
- How often do I measure?

TOOLS

- monitoring

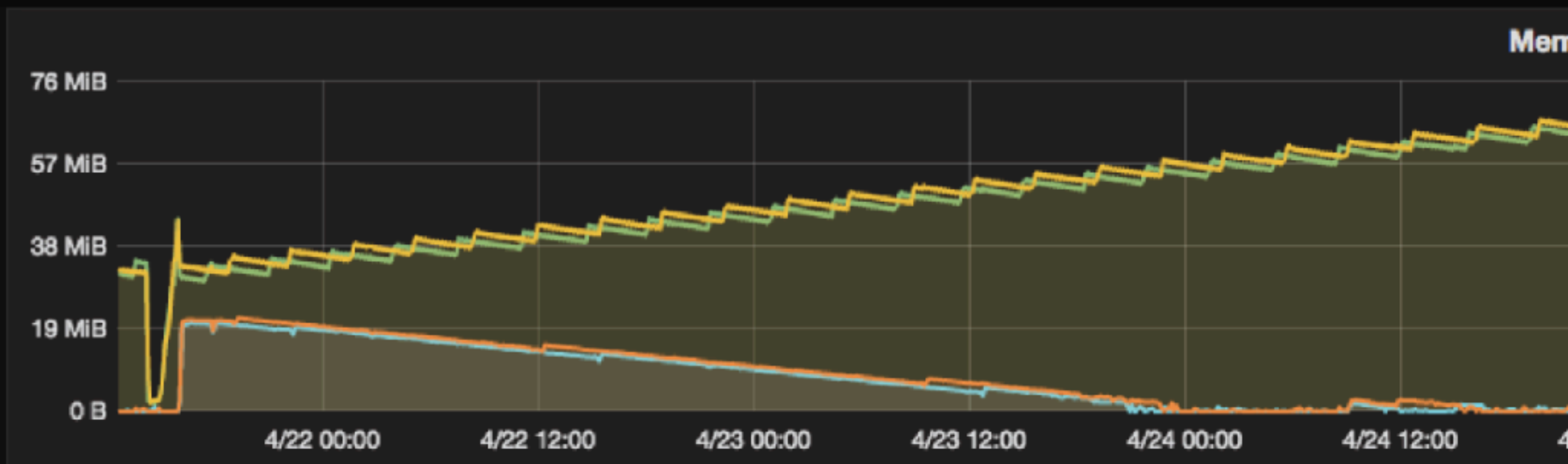


- load testing

e.g. go-wrk, k6, pewpew, jmeter, gatling+taurus

- benchmarking
- profiling
- visualising

/metrics



```
{"BadRequests":{"count":0},"IdentifyAccountTime":  
{"15m.rate":20.090308331759587,"1m.rate":18.385546387864803,"5m.rate":20.155648494583627,"  
75%":21404.75,"95%":33994.499999999985,"99%":200137.260000000164,"99.9%":1128821.51700000014  
,"count":1100,"max":1140141,"mean":24179.230544747083,"mean.rate":21.99810286996967,"media  
n":16504,"min":9682,"stddev":56887.9633075614},"LocatorTime.locator-kat-dev-vm-gbbr-  
brightpearl-com:80/":  
{"15m.rate":82.40710272987735,"1m.rate":50.152064088948514,"5m.rate":76.7386188585391,"75%  
":758099801.25,"95%":1068668907.3,"99%":2107370126.4000008,"99.9%":3768931930.796001,"coun  
t":1100,"max":3775973046,"mean":569778968.9484435,"mean.rate":24.453846552291488,"median":  
508510906.5,"min":203897682,"stddev":382597027.50912994},"PageFetchingTime":  
{"15m.rate":1.1734873744321592,"1m.rate":9.99241474280071,"5m.rate":3.2456288979497447,"75  
%":1436500919.5,"95%":2605897930.5,"99%":4922161345.37,"99.9%":4940945331.111,"count":1100  
,"max":4940955462,"mean":901225883.2140077,"mean.rate":21.998101373787847,"median":1833198  
,"min":362473,"stddev":1113924304.429029},"RequestTime":  
{"15m.rate":1.1734873744321592,"1m.rate":9.99241474280071,"5m.rate":3.2456288979497447,"75  
%":2043938599.25,"95%":3609776337.6499996,"99%":5445697496.38,"99.9%":5485534175.186,"coun  
t":1100,"max":5485572286,"mean":1460402107.6614785,"mean.rate":21.99810256771803,"median"  
:1281296551,"min":204602185,"stddev":1209669851.3536835},"RequestsReceived":  
{"count":1100},"SDTime":  
{"15m.rate":1.172109877516385,"1m.rate":9.811960484865835,"5m.rate":3.234157671839152,"75%  
":746222535,"95%":1080152863.5499997,"99%":2110097145.4500003,"99.9%":3768957463.677001,"c  
ount":1100,"max":3775998237,"mean":569384948.4688716,"mean.rate":21.998101806233205,"media  
n":507778172,"min":203917982,"stddev":389814216.44867057},"SessionDBTime":  
{"15m.rate":0,"1m.rate":0,"5m.rate":0,"75%":0,"95%":0,"99%":0,"99.9%":0,"count":0,"max":0,  
"mean":0,"mean.rate":0,"median":0,"min":0,"stddev":0}}
```

DIAGNOSTICS
BEFORE
DIAGNOSIS

INSTALL A PROFILER



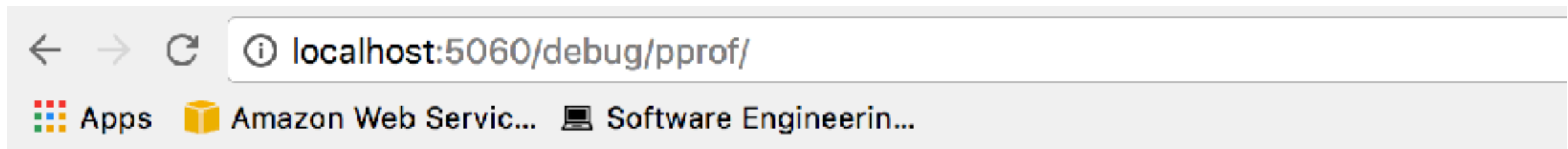
```
import _ "net/http/pprof"
```



```
http.ListenAndServe(":5060", nil)
```



```
/debug/pprof/
```



/debug/pprof/

profiles:

0 [block](#)

7 [goroutine](#)

0 [heap](#)

0 [mutex](#)

7 [threadcreate](#)

[full goroutine stack dump](#)

GENERATE SOME LOAD



```
go get github.com/adjust/go-wrk
```



```
go-wrk -n 15000 http://localhost:5060/metrics
```



```
go tool pprof --seconds 5
```

```
http://localhost: 5060/debug/pprof/profile
```

× ~ (bash)

kat at **Kats-laptop** in ~

```
$ go-wrk -n 15000 http://localhost:5060/metrics
```

× ~ (bash)

kat at **Kats-laptop** in ~

```
$ go tool pprof --seconds 5 http://localhost:5060/debug/pprof/profile
```



```
$ go-wrk -n 15000 http://localhost:5060/metrics
```

```
=====BENCHMARK=====
```

```
URL: http://localhost:5060/metrics
```

```
Used Connections: 100
Used Threads: 1
Total number of calls: 15000
```

```
=====TIMINGS=====
```

```
Total time passed: 76.14s
Avg time per request: 59.51ms
Requests per second: 197.00
Median time per request: 60.35ms
99th percentile time: 83.81ms
Slowest time for request: 95.00ms
```

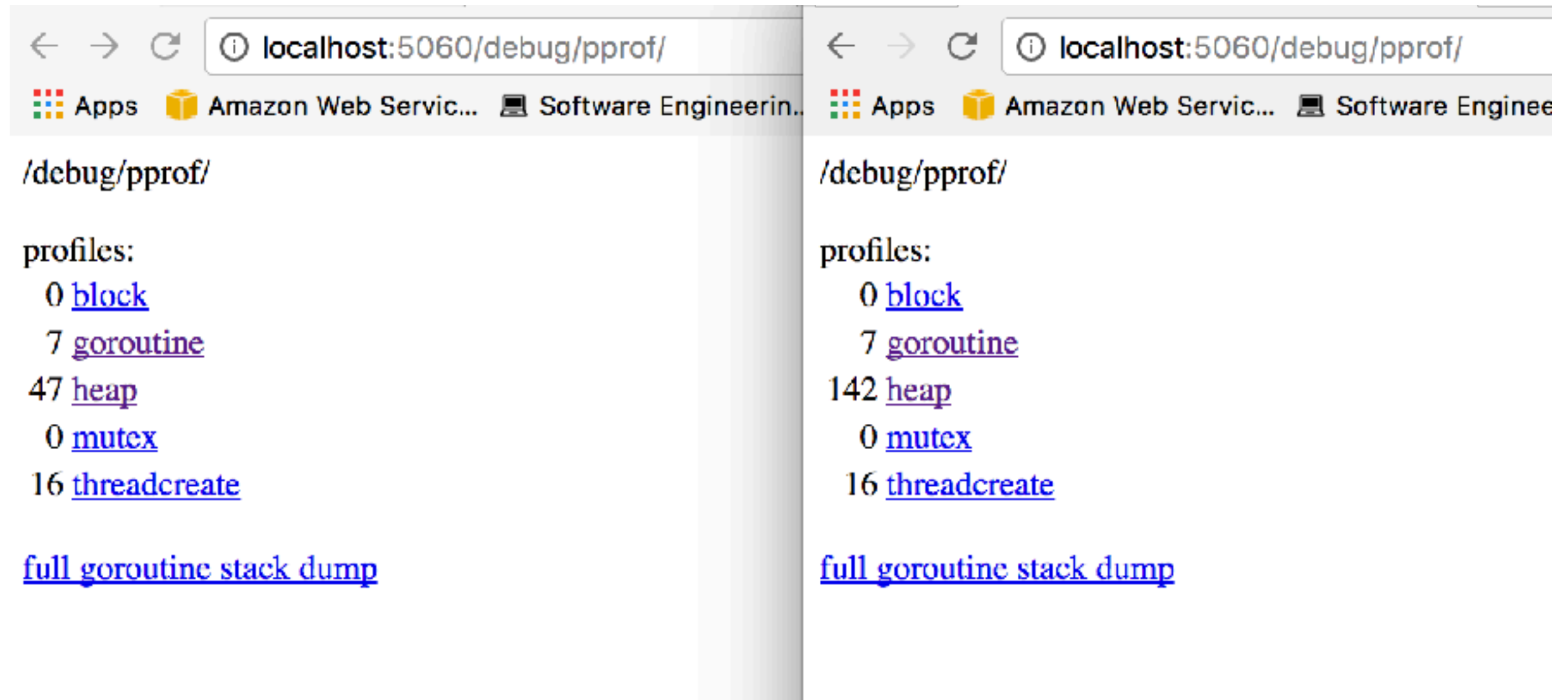
```
=====DATA=====
```

```
Total response body sizes: 29322336
Avg response body per request: 1954.82ms
Transfer rate per second: 385091.26 Byte/s (0.39 MByte/s)
```

```
=====RESPONSES=====
```

```
20X Responses: 15000 (100.00%)
30X Responses: 0 (0.00%)
40X Responses: 0 (0.00%)
50X Responses: 0 (0.00%)
Errors: 0 (0.00%)
```

/debug/pprof



/debug/pprof/heap

```
1: 48 [1: 48] @ 0x14099b4 0x140b13e 0x140adb7 0x1410b8f 0x141b6f7 0x13f40cf 0x13f3aa6 0x1423653 0x12722e4 0x126e4dd 0
# 0x14099b3 git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics.NewEWMA+0x83 /U
kat/Code/go-work/src/git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics/ewma.go:23
# 0x14099b3 git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics.NewEWMA5+0x83 /U
kat/Code/go-work/src/git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics/ewma.go:33
# 0x140b13d git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics.newStandardMeter+0x3
Users/kat/Code/go-work/src/git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics/meter.go:1
# 0x140adb6 git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics.NewMeter+0x46 /U
kat/Code/go-work/src/git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics/meter.go:34
# 0x1410b8e git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics.NewTimer+0x8e /U
kat/Code/go-work/src/git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics/timer.go:67
# 0x141b6f6 git.brightpearl.com/orbs/front-controller/metrics.(*Handler).AddLocatorTime+0x106 /Users/kat
work/src/
...
```

```
1: 48 [1: 48] @ 0x1409a64 0x140b156 0x140adb7 0x1410b8f 0x141b6f7 0x13f40cf 0x13f3aa6 0x1423653 0x12722e4 0x126e4dd 0
# 0x1409a63 git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics.NewEWMA+0x83 /U
kat/Code/go-work/src/git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics/ewma.go:23
# 0x1409a63 git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics.NewEWMA15+0x83 /U
kat/Code/go-work/src/git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics/ewma.go:38
# 0x140b155 git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics.newStandardMeter+0x5
Users/kat/Code/go-work/src/git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics/meter.go:1
# 0x140adb6 git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics.NewMeter+0x46 /U
kat/Code/go-work/src/git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics/meter.go:34
# 0x1410b8e git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics.NewTimer+0x8e /U
kat/Code/go-work/src/git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics/timer.go:67
# 0x141b6f6 git.brightpearl.com/orbs/front-controller/metrics.(*Handler).AddLocatorTime+0x106 /Users/kat
work/src/git.brightpearl.com/orbs/front-controller/metrics/metrics.go:117
...
```

```
/debug/pprof/heap
```

```
1: 48 [1: 48] @ 0x14099b4 0x140b13e 0x140adb7 0x1410b8f 0x141b6f7 0x13f40cf 0x13f3aa6 0x1423653 0x12722c4 0x126c4d1 0  
#   0x14099b3      git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics.NewEWMA+0x83    /U  
kat/Code/go-work/src/git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-me-trics/ewma.go:23  
#   0x14099b3      git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metric...NewEWMA5+0x83    /U  
kat/Code/go-work/src/git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-me-trics/ewma.go:33  
#   0x140b13d      git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metric...newStandardMeter+0x3    /U  
Users/kat/Code/go-work/src/git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics/meter.go:1  
#   0x140adb6      git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics.NewMeter+0x46    /U  
kat/Code/go-work/src/git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics/meter.go:34  
#   0x1410b8e      git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics.NewTimer+0x8e    /U  
kat/Code/go-work/src/git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics/timer.go:67  
#   0x141b6f6      git.brightpearl.com/orbs/front-controller/metrics.(*Handler).AddLocatorTime+0x106            /Users/kat  
work/src/  
...
```

```
1: 48 [1: 48] @ 0x1409a64 0x140b156 0x140adb7 0x1410b8f 0x141b6f7 0x13f40cf 0x13f3aa6 0x1423652 0x12722e4 0x126c1d0 0x126c1d0
# 0x1409a63 git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics.NewEWMA+0x83 /Users/kat/Code/go-work/src/git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics/ewma.go:23
# 0x1409a63 git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics.NewEWMA15+0x83 /Users/kat/Code/go-work/src/git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics/ewma.go:38
# 0x140b155 git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics.newStandardMeter+0x5 /Users/kat/Code/go-work/src/git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics/meter.go:1
# 0x140adb6 git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics.NewMeter+0x46 /Users/kat/Code/go-work/src/git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics/meter.go:34
# 0x1410b8e git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics.NewTimer+0x8e /Users/kat/Code/go-work/src/git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics/timer.go:67
# 0x141b6f6 git.brightpearl.com/orbs/front-controller/metrics.(*Handler).AddLocatorTime+0x106 /Users/kat/Code/go-work/src/git.brightpearl.com/orbs/front-controller/metrics/metrics.go:117
...
```

```
/debug/pprof/heap
```

[illegible]

/debug/pprof/heap



```
wget -O heap.prof http://localhost:5060/debug/pprof/heap
```



```
go tool pprof --inuse_objects heap.prof
```

```
(pprof) top10
Showing nodes accounting for 24107, 100% of 24107 total
Showing top 10 nodes out of 35
flat flat% sum% cum cum%
10923 45.31% 45.31% 10923 45.31% git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics.NewEWMA /
ndor/github.com/rcrowley/go-metrics/ewma.go (inline)
5461 22.65% 67.96% 5461 22.65% net/http.(*Transport).dialConn /Users/kat/goroots/go1.9/src/net/http/transport.go
4096 16.99% 84.95% 4096 16.99% net.newFD /Users/kat/goroots/go1.9/src/net/fd_unix.go
2731 11.33% 96.28% 2731 11.33% runtime.malg /Users/kat/goroots/go1.9/src/runtime/proc.go
512 2.12% 98.41% 512 2.12% net/http.(*ServeMux).Handle /Users/kat/goroots/go1.9/src/net/http/server.go
384 1.59% 100% 384 1.59% bufio.NewWriterSize /Users/kat/goroots/go1.9/src/bufio/bufio.go
0 0% 100% 512 2.12% git.brightpearl.com/orbs/front-controller/api.NewInternalServer /Users/kat/Code/go-work/sr
0 0% 100% 10923 45.31% git.brightpearl.com/orbs/front-controller/metrics.(*Handler).AddLocatorTime /Users/kat/Cod
s.go
0 0% 100% 10923 45.31% git.brightpearl.com/orbs/front-controller/proxy.(*Handler).ServeHTTP /Users/kat/Code/go-w
```

```
(pprof) top10
Showing nodes accounting for 24107, 100% of 24107 total
Showing top 10 nodes out of 35
      flat flat% sum%      cum cum%
10923 45.31% 45.31% 10923 45.31% git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-metrics.NewEWMA /U
ndor/github.com/rcrowley/go-metrics/ewma.go (inline)
 5461 22.65% 67.96%  5461 22.65% net/http.(*Transport).dialConn /Users/kat/goroots/go1.9/src/net/http/transport.go
 4096 16.99% 84.95%  4096 16.99% net.newFD /Users/kat/goroots/go1.9/src/net/fd_unix.go
 2731 11.33% 96.28%  2731 11.33% runtime.malg /Users/kat/goroots/go1.9/src/runtime/proc.go
  512  2.12% 98.41%   512  2.12% net/http.(*ServeMux).Handle /Users/kat/goroots/go1.9/src/net/http/server.go
  384  1.59% 100%     384  1.59% bufio.NewWriterSize /Users/kat/goroots/go1.9/src/bufio/bufio.go
    0  0% 100%     512  2.12% git.brightpearl.com/orbs/front-controller/api.NewInternalServer /Users/kat/Code/go-work/sr
    0  0% 100% 10923 45.31% git.brightpearl.com/orbs/front-controller/metrics.(*Handler).AddLocatorTime /Users/kat/Cod
s.go
    0  0% 100% 10923 45.31% git.brightpearl.com/orbs/front-controller/proxy.(*Handler).ServeHTTP /Users/kat/Code/go-w
```


BENCHMARKS

```
func (m *Handler) AddLocatorTime(host string, d time.Duration) {  
    metrics.GetOrRegister(fmt.Sprintf("LocatorTime.%s",  
escapeHost(host)), metrics.NewTimer()).(metrics.Timer).Update(d)  
}
```

```
func BenchmarkAddLocatorTime(b *testing.B) {  
    h := NewHandler(false, time.Minute)  
    host := "some-epic.host"  
    b.ReportAllocs()  
    for i := 0; i < b.N; i++ {  
        h.AddLocatorTime(host, time.Minute)  
    }  
}
```

```
$ go test -bench=. .
```

```
goos: darwin
```

```
goarch: amd64
```

```
pkg: git.brightpearl.com/orbs/front-controller/metrics
```

```
BenchmarkAddLocatorTime-8      300000      5870 ns/op    19056 B/op    14 allocs/op
```

```
PASS
```

```
ok  git.brightpearl.com/orbs/front-controller/metrics    1.960s
```

go rcrowley metrics GetOrRegister timer memory issue



All

Shopping

Images

News

Videos

More

Settings

Tools

About 48 results (0.56 seconds)

Metrics library leaks memory · Issue #897 · Shopify/sarama · GitHub

<https://github.com/Shopify/sarama/issues/897> ▼

Jun 19, 2017 - Note that we do not use **timer metrics** but that would result in the same **issue** as the ...

Close, May Cause **Memory Leak** to **Metrics** library leaks **memory** 23 days ago ... 912.56MB 96.10%

github.com/lday0321/myProj/vendor/github.com/rcrowley/go-metrics. ... **GetOrRegister** 0 0% 98.89%

- func (t *StandardTimer) Time(f func())
- func (t *StandardTimer) Update(d time.Duration)
- func (t *StandardTimer) UpdateSince(ts time.Time)
- func (t *StandardTimer) Variance() float64

type Timer

- func GetOrRegisterTimer(name string, r Registry) Timer
- func NewCustomTimer(h Histogram, m Meter) Timer
- func NewRegisteredTimer(name string, r Registry) Timer
- func NewTimer() Timer

type TimerSnapshot

- func (t *TimerSnapshot) Count() int64

```
func (m *Handler) AddLocatorTime(host string, d time.Duration) {  
    metrics.GetOrRegisterTimer(fmt.Sprintf("LocatorTime.%s",  
escapeHost(host)), nil).Update(d)  
}
```

```
$ go test -bench=. .
```

```
goos: darwin
```

```
goarch: amd64
```

```
pkg: git.brightpearl.com/orbs/front-controller/metrics
```

```
BenchmarkAddLocatorTime-8      1000000      1011 ns/op      80 B/op
```

```
PASS
```

```
ok  git.brightpearl.com/orbs/front-controller/metrics  1.039s
```

4 allocs/op

```

(pprof) top10
Showing nodes accounting for 3623.35kB, 100% of 3623.35kB total
Showing top 10 nodes out of 16
      flat flat% sum%        cum    cum%   bufio.NewWriterSize /Users/kat/goroots/go1.9/src/bufio/bufio.go (inline)
2055.01kB 56.72% 56.72% 2055.01kB 56.72%
1056.33kB 29.15% 85.87% 1056.33kB 29.15%  regexp.(*bitState).reset /Users/kat/goroots/go1.9/src/regexp/backtrack.go
 512.01kB 14.13% 100%   512.01kB 14.13%  net/http/httputil.cloneHeader /Users/kat/goroots/go1.9/src/net/http/httputil/reverseproxy.go
      0      0% 100%      514kB 14.19%  bufio.NewWriter /Users/kat/goroots/go1.9/src/bufio/bufio.go (inline)
      0      0% 100% 1568.34kB 43.28%  git.brightpearl.com/orbs/front-controller/proxy.(*Handler).ServeHTTP /Users/kat/Code/go-work/src
      0      0% 100%      514kB 14.19%  net/http.(*Transport).dialConn /Users/kat/goroots/go1.9/src/net/http/transport.go
      0      0% 100%      514kB 14.19%  net/http.(*Transport).getConn.func4 /Users/kat/goroots/go1.9/src/net/http/transport.go
      0      0% 100%      513kB 14.16%  net/http.(*conn).readRequest /Users/kat/goroots/go1.9/src/net/http/server.go
      0      0% 100% 3109.34kB 85.81%  net/http.(*conn).serve /Users/kat/goroots/go1.9/src/net/http/server.go
      0      0% 100%   1541kB 42.53%  net/http.newBufioWriterSize /Users/kat/goroots/go1.9/src/net/http/server.go
(pprof)

```

kill -6



SIGABRT: abort
PC=0x105e193 m=0 sigcode=0

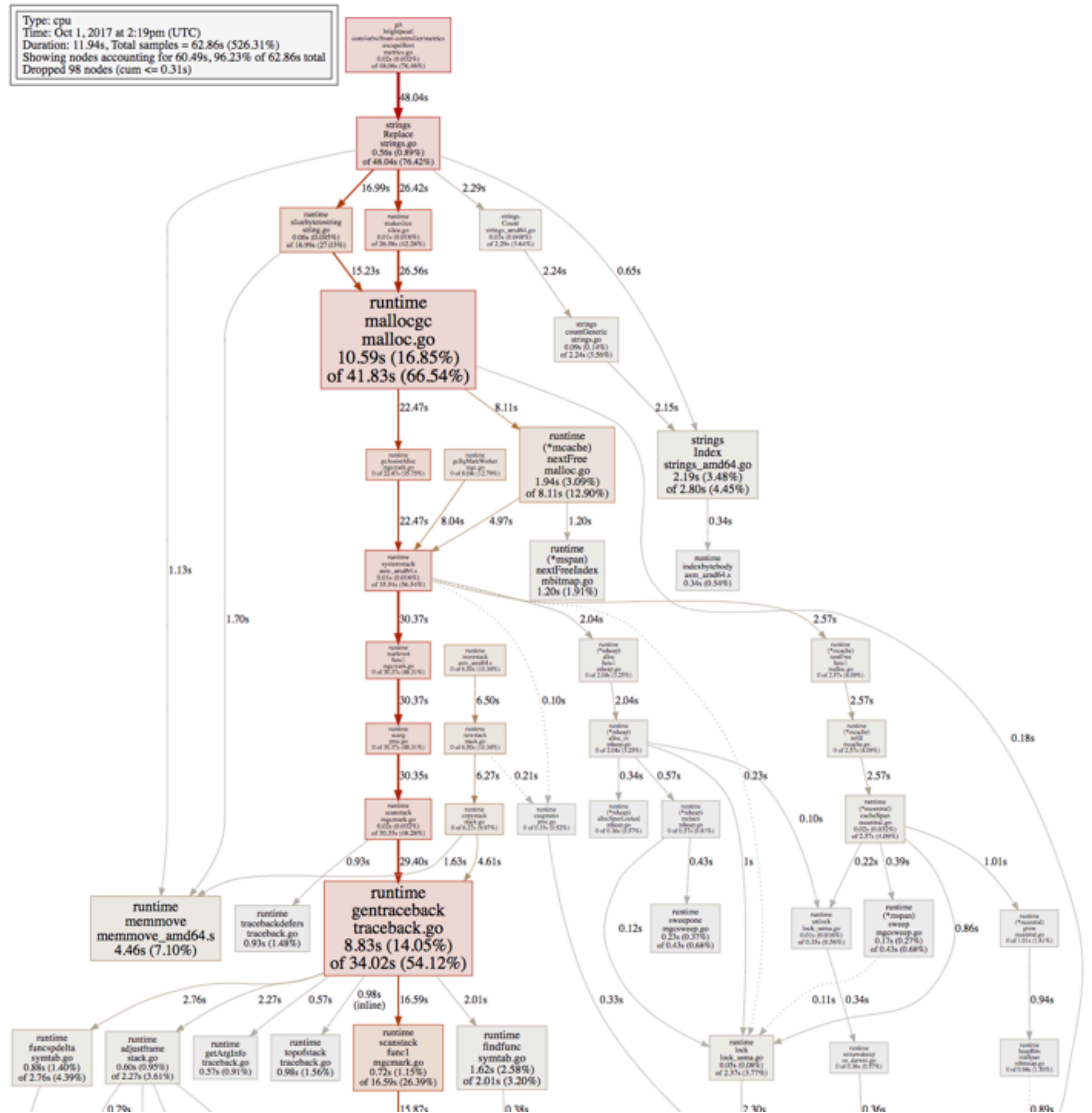
goroutine 0 [idle]:
runtime.kevent(0x7fff00000005, 0x0, 0x0, 0x7fff5fbfe4f0, 0x40, 0x0, 0xffffffffc,
0xffffffffc00000000, 0x0, 0x0, ...)
...

goroutine 1 [IO wait]:
internal/poll.runtime_pollWait(0x197f1f8, 0x72, 0xffffffffffffffff)
 /Users/kat/goroots/go1.9/src/runtime/netpoll.go:173 +0x57
...

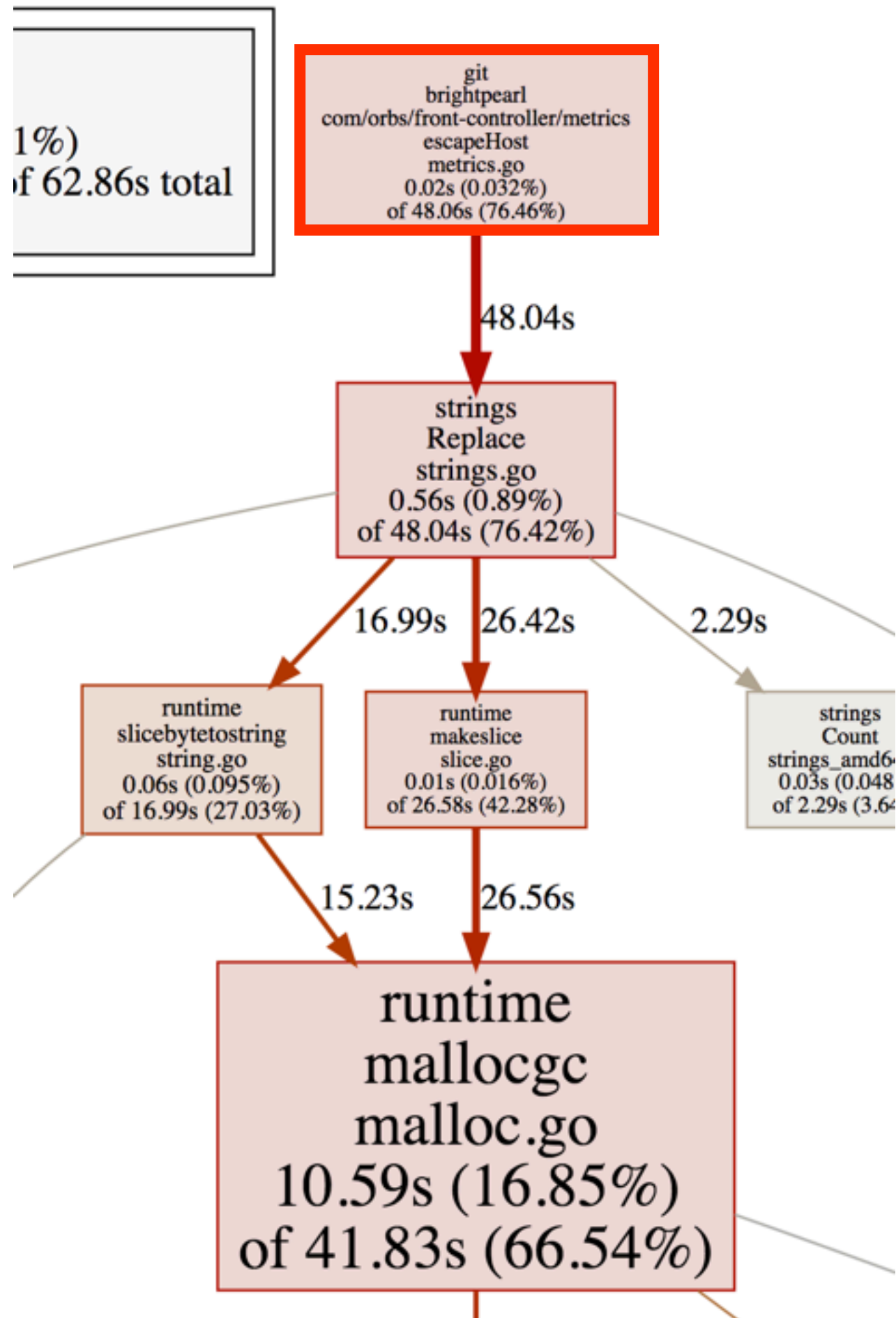
goroutine 7 [chan receive]:
git.brightpearl.com/orbs/front-controller/vendor/github.com/rcrowley/go-
metrics.(*meterArbiter).tick(0x17f56e0)
...

rax	0x4	r11	0x246
rbx	0x0	r12	0x0
rcx	0x7fff5fbfe480	r13	0xf7
rdx	0x0	r14	0x77
rdi	0x5	r15	0x0
rsi	0x0	rip	0x105e193
rbp	0x7fff5fbfecf0	rflags	0x247
rsp	0x7fff5fbfe480	cs	0x7
r8	0x40	fs	0x0
r9	0x0	gs	0x0
r10	0x7fff5fbfe4f0		

(pprof) web



(pprof) web



```
func escapeHost(host string) string {  
    escaped := strings.Replace(host, ".", "-", -1)  
    escaped = strings.Replace(escaped, "http://", "", -1)  
    escaped = strings.Replace(escaped, "https://", "", -1)  
  
    return escaped  
}
```

```
func BenchmarkEscapeHost(b *testing.B) {
```

```
    testCases := []struct {
```

```
        name, host string
```

```
    }{
```

```
        {
```

```
            "empty",
```

```
            "",
```

```
        },
```

```
        {
```

```
            "http dot",
```

```
            "http://foobar.com",
```

```
        },
```

```
        {
```

```
            "https dot",
```

```
            "https://foobar.com",
```

```
        },
```

```
        {
```

```
            "http hyphen",
```

```
            "http://foobar-com",
```

```
        },
```

```
        {
```

```
            "https hyphen",
```

```
            "https://foobar-com",
```

```
        },
```

```
        {
```

```
            "1hyphen",
```

```
            "foobar-com",
```

```
        },
```

```
        {
```

```
            "2hyphen",
```

```
            "stg-carbon-1",
```

```
        },
```

```
        {
```

```
            "1dot",
```

```
            "foobar.com",
```

```
        },
```

```
    {
```

```
        "2dot",
```

```
        "foo.bar.com",
```

```
    },
```

```
    {
```

```
        "mixed",
```

```
        "foo-bar.com",
```

```
    },
```

```
    {
```

```
        "full",
```

```
        "ec2-54-188-109-96.us-
```

```
west-2.compute.amazonaws.com",
```

```
    },
```

```
    {
```

```
        "weird",
```

```
        "http://ec2-54-188-109-96.us-
```

```
west-2.compute.amazonaws.com",
```

```
    },
```

```
    {
```

```
        "prefix",
```

```
        ".euw1",
```

```
    },
```

```
    {
```

```
        "suffix",
```

```
        "use2.",
```

```
    },
```

```
    {
```

```
        "none",
```

```
        "stg",
```

```
    },
```

```
    }
```

```
    for _, tc := range testCases {
```

```
        b.Run(tc.name, func(b *testing.B) {
```

```
            b.ReportAllocs()
```

```
            b.ResetTimer()
```

```
            for i := 0; i < b.N; i++ {
```

```
                escapeHost(tc.host)
```

```
            }
```

```
        })
```

```
    }
```

```
}
```

go test -cpuprofile cpu.prof -memprofile mem.prof -bench .

\$ go test -cpuprofile cpu.prof -memprofile mem.prof -bench . > baseline.txt

goos: darwin

goarch: amd64

BenchmarkEscapeHost/empty-8	50000000	34.9 ns/op	0 B/op	0 allocs/op
BenchmarkEscapeHost/http_dot-8	10000000	212 ns/op	96 B/op	4 allocs/op
BenchmarkEscapeHost/https_dot-8	10000000	233 ns/op	96 B/op	4 allocs/op
BenchmarkEscapeHost/http_hyphen-8	10000000	142 ns/op	32 B/op	2 allocs/op
BenchmarkEscapeHost/https_hyphen-8	10000000	133 ns/op	32 B/op	2 allocs/op
BenchmarkEscapeHost/1hyphen-8	30000000	53.3 ns/op	0 B/op	0 allocs/op
BenchmarkEscapeHost/2hyphen-8	30000000	56.8 ns/op	0 B/op	0 allocs/op
BenchmarkEscapeHost/1dot-8	10000000	124 ns/op	32 B/op	2 allocs/op
BenchmarkEscapeHost/2dot-8	10000000	156 ns/op	32 B/op	2 allocs/op
BenchmarkEscapeHost/mixed-8	10000000	142 ns/op	32 B/op	2 allocs/op
BenchmarkEscapeHost/full-8	10000000	219 ns/op	128 B/op	2 allocs/op
BenchmarkEscapeHost/weird-8	3000000	351 ns/op	256 B/op	4 allocs/op
BenchmarkEscapeHost/prefix-8	20000000	111 ns/op	10 B/op	2 allocs/op
BenchmarkEscapeHost/suffix-8	20000000	100 ns/op	10 B/op	2 allocs/op
BenchmarkEscapeHost/none-8	50000000	38.1 ns/op	0 B/op	0 allocs/op

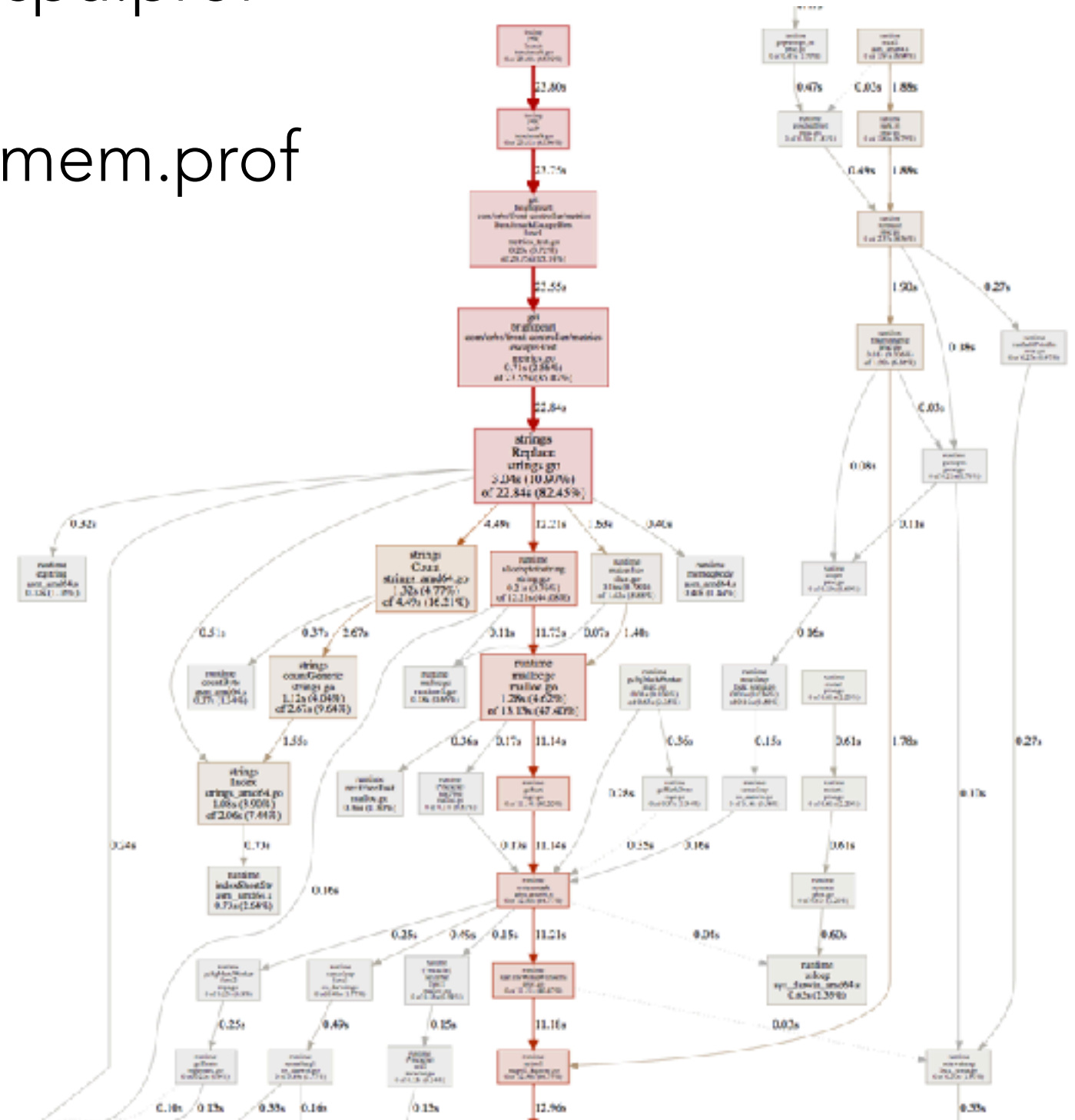
PASS

ok git.brightpearl.com/orbs/front-controller/metrics 28.308s

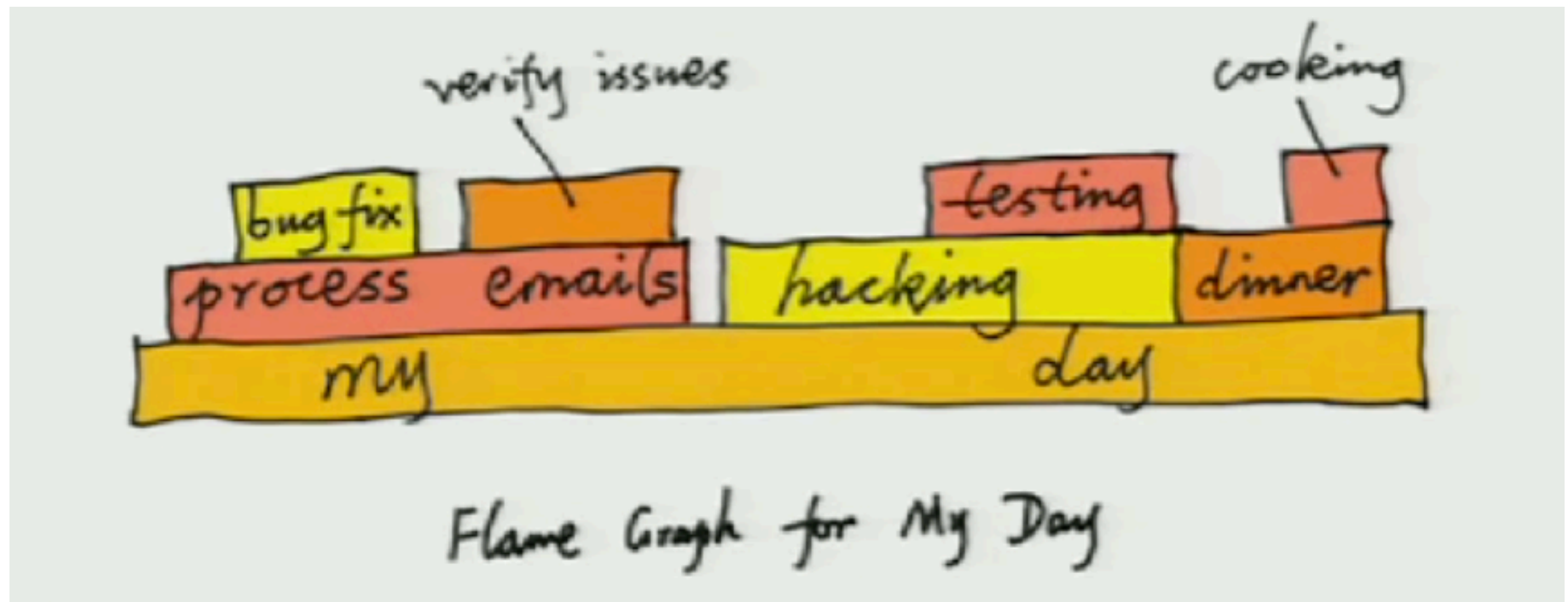
```
go tool pprof metrics.test cpu.prof
```

```
go tool pprof metrics.test mem.prof
```

👉 (pprof) web



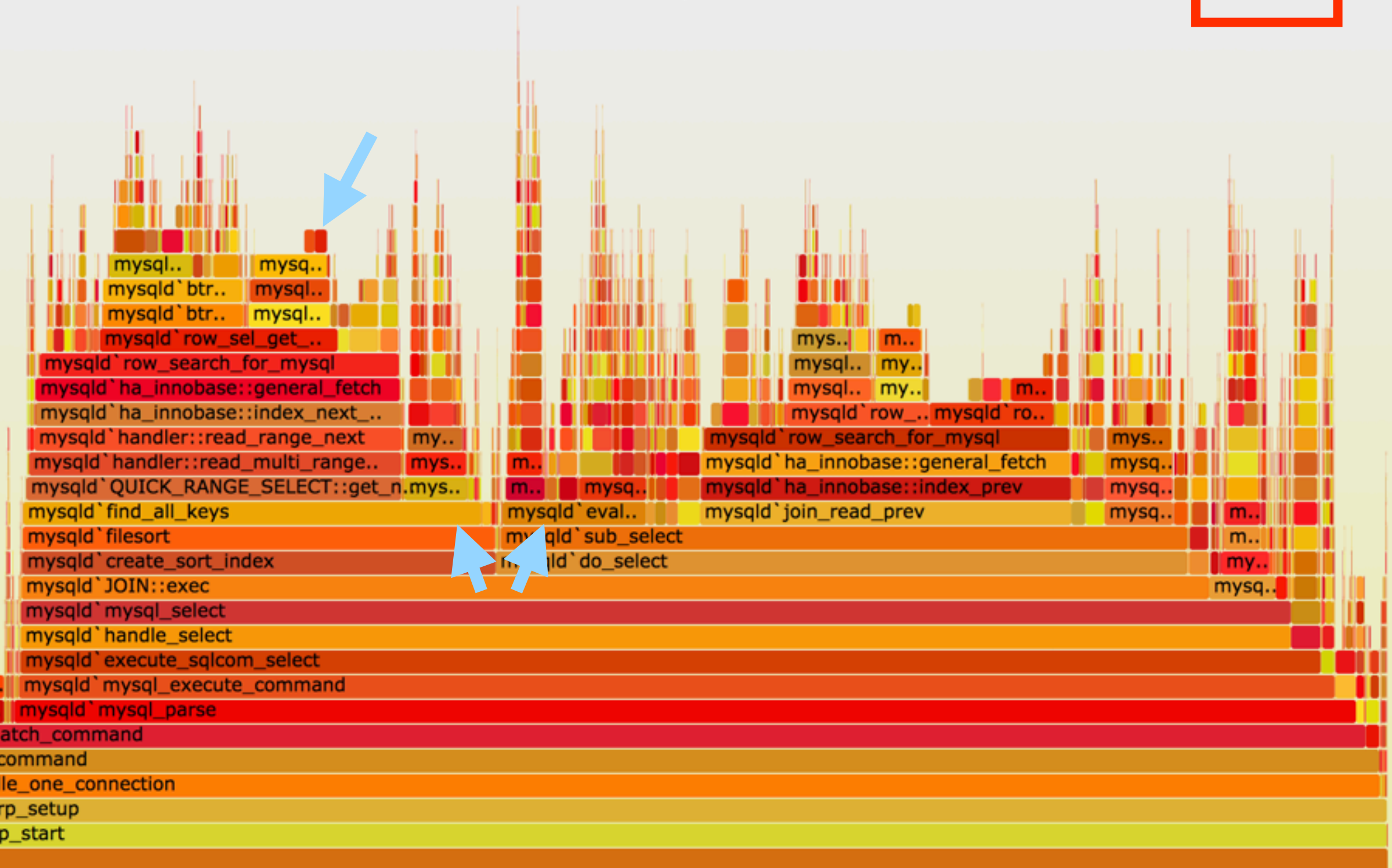
FLAME GRAPHS



source: Yichun Zhang

see: <https://www.youtube.com/watch?v=nZfNehCzGdw> and <http://www.brendangregg.com/flamegraphs.html>

Flame Graph

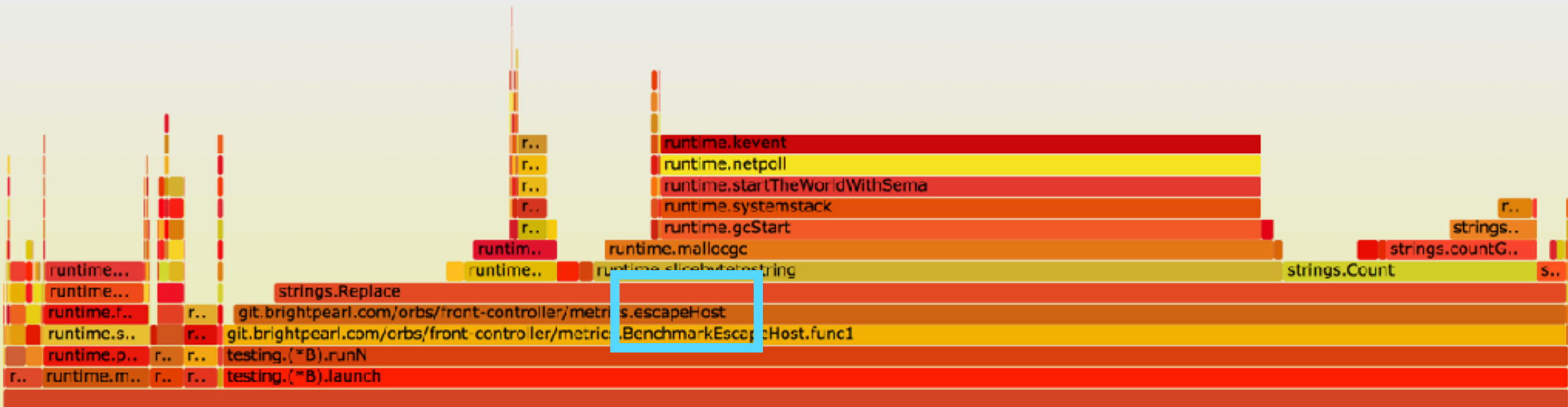


👉 go get github.com/uber/go-torch
(+ README instructions)

✌ go-torch metrics.test cpu.prof

👉 go-torch --seconds 5 <http://XXX/debug/pprof/profile>

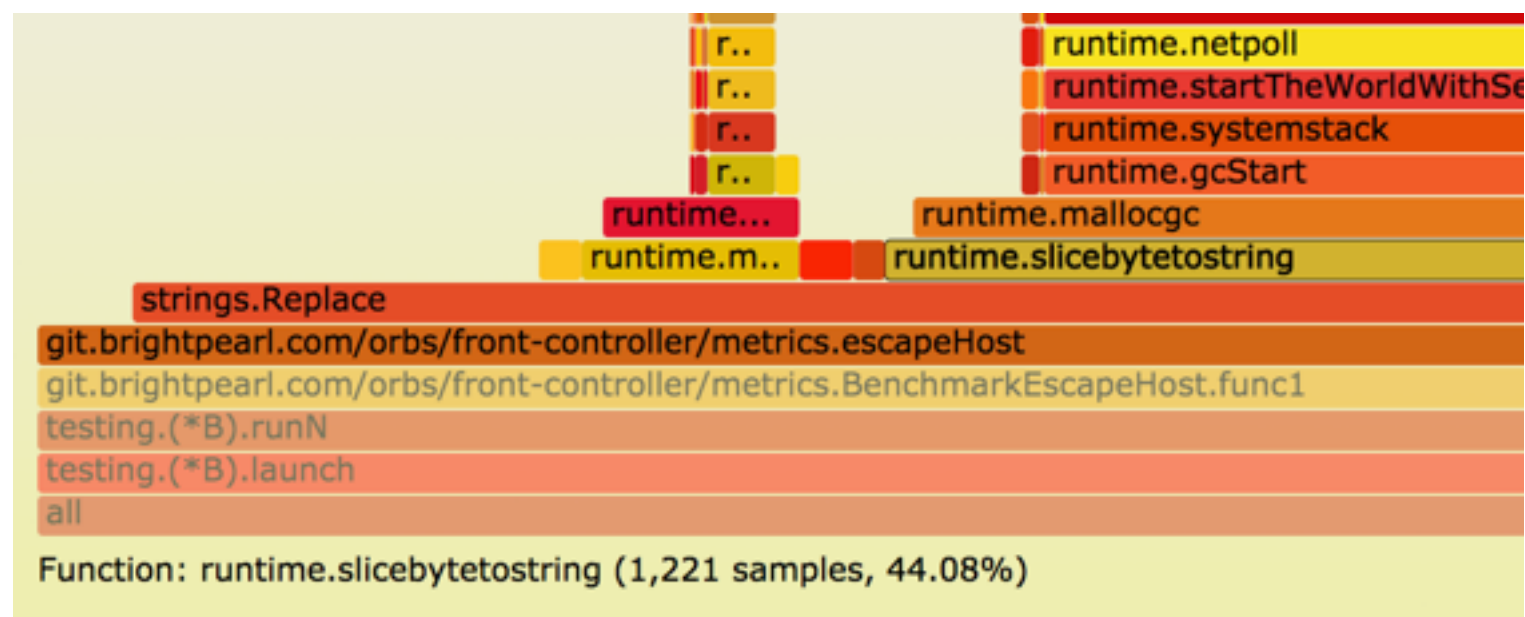
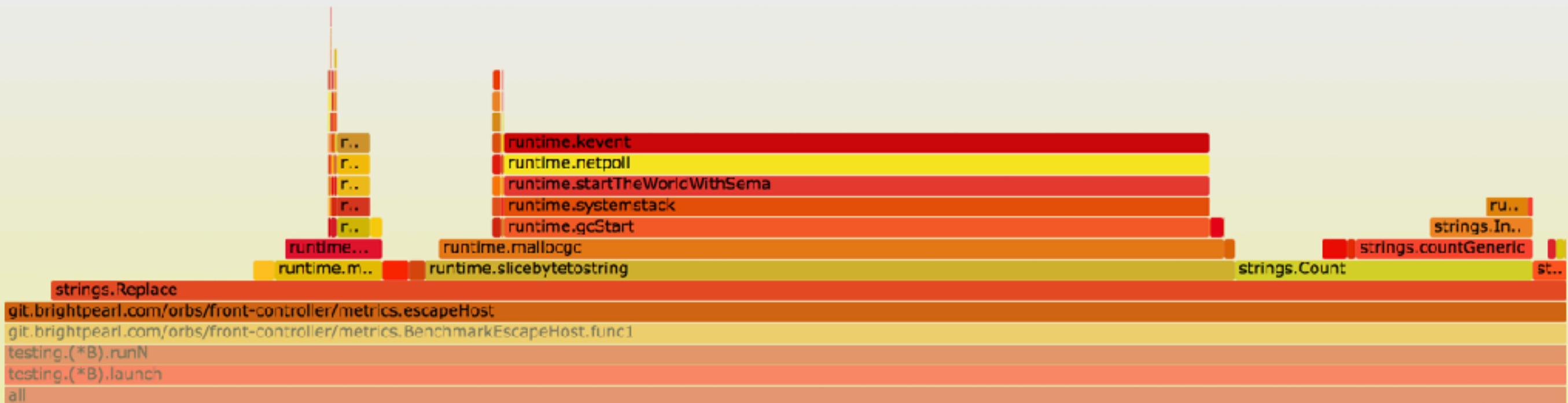
👉 open torch.svg



Reset Zoom

Flame Graph

Search



```
func escapeHost(host string) string {  
    escaped := strings.Replace(host, ".", "-", -1)  
    escaped = strings.Replace(escaped, "http://", "", -1)  
    escaped = strings.Replace(escaped, "https://", "", -1)  
  
    return escaped  
}
```



```
func escapeHost(host string) string {  
    escaped := bytes.Replace([]byte(host), []byte("."), []byte("-"), -1)  
    escaped = bytes.Replace(escaped, []byte("http://"), nil, -1)  
    escaped = bytes.Replace(escaped, []byte("https://"), nil, -1)  
  
    return string(escaped)  
}
```

```
$ benchcmp baseline.txt swap-for-byte.txt
```

benchmark	old ns/op	new ns/op	delta
BenchmarkEscapeHost/empty-8	34.9	105	+200.86%
BenchmarkEscapeHost/http_dot-8	212	323	+52.36%
BenchmarkEscapeHost/https_dot-8	233	320	+37.34%
BenchmarkEscapeHost/http_hyphen-8	142	297	+109.15%
BenchmarkEscapeHost/https_hyphen-8	133	308	+131.58%
BenchmarkEscapeHost/1hyphen-8	53.3	264	+395.31%
BenchmarkEscapeHost/2hyphen-8	56.8	256	+350.70%
BenchmarkEscapeHost/1dot-8	124	263	+112.10%
BenchmarkEscapeHost/2dot-8	156	282	+80.77%
BenchmarkEscapeHost/mixed-8	142	264	+85.92%
BenchmarkEscapeHost/full-8	219	394	+79.91%
BenchmarkEscapeHost/weird-8	351	428	+21.94%
BenchmarkEscapeHost/prefix-8	111	239	+115.32%
BenchmarkEscapeHost/suffix-8	100	227	+127.00%
BenchmarkEscapeHost/none-8	38.1	215	+464.30%

benchmark	old allocs	new allocs	delta
BenchmarkEscapeHost/empty-8	0	0	+0.00%
BenchmarkEscapeHost/http_dot-8	4	5	+25.00%
BenchmarkEscapeHost/https_dot-8	4	5	+25.00%
BenchmarkEscapeHost/http_hyphen-8	2	5	+150.00%
BenchmarkEscapeHost/https_hyphen-8	2	5	+150.00%
BenchmarkEscapeHost/1hyphen-8	0	5	+Inf%
BenchmarkEscapeHost/2hyphen-8	0	5	+Inf%
BenchmarkEscapeHost/1dot-8	2	5	+150.00%
BenchmarkEscapeHost/2dot-8	2	5	+150.00%
BenchmarkEscapeHost/mixed-8	2	5	+150.00%
BenchmarkEscapeHost/full-8	2	5	+150.00%
BenchmarkEscapeHost/weird-8	4	5	+25.00%
BenchmarkEscapeHost/prefix-8	2	5	+150.00%
BenchmarkEscapeHost/suffix-8	2	5	+150.00%
BenchmarkEscapeHost/none-8	0	5	+Inf%

benchmark	old bytes	new bytes	delta
BenchmarkEscapeHost/empty-8	0	0	+0.00%
BenchmarkEscapeHost/http_dot-8	96	112	+16.67%
BenchmarkEscapeHost/https_dot-8	96	128	+33.33%
BenchmarkEscapeHost/http_hyphen-8	32	112	+250.00%
BenchmarkEscapeHost/https_hyphen-8	32	128	+300.00%
BenchmarkEscapeHost/1hyphen-8	0	80	+Inf%
BenchmarkEscapeHost/2hyphen-8	0	80	+Inf%
BenchmarkEscapeHost/1dot-8	32	80	+150.00%
BenchmarkEscapeHost/2dot-8	32	80	+150.00%
BenchmarkEscapeHost/mixed-8	32	80	+150.00%
BenchmarkEscapeHost/full-8	128	320	+150.00%
BenchmarkEscapeHost/weird-8	256	320	+25.00%
BenchmarkEscapeHost/prefix-8	10	40	+300.00%
BenchmarkEscapeHost/suffix-8	10	40	+300.00%
BenchmarkEscapeHost/none-8	0	40	+Inf%

```
func escapeHost(host string) string {  
    escaped := bytes.Replace([]byte(host), []byte("."), []byte("-"), -1)  
    escaped = bytes.Replace(escaped, []byte("http://"), nil, -1)  
    escaped = bytes.Replace(escaped, []byte("https://"), nil, -1)  
  
    return string(escaped)  
}
```



```
func escapeHost(host string) string {  
    escaped := bytes.Replace([]byte(host), []byte("."), []byte("-"), -1)  
    escaped = bytes.TrimPrefix(escaped, []byte("http://"))  
    escaped = bytes.TrimPrefix(escaped, []byte("https://"))  
  
    return string(escaped)  
}
```



```
$ benchcmp baseline.txt prefix.txt
```

benchmark	old ns/op	new ns/cp	delta
BenchmarkEscapeHost/empty-8	34.9	65.8	+88.54%
BenchmarkEscapeHost/http_dot-8	212	178	-16.04%
BenchmarkEscapeHost/https_dot-8	233	183	-21.46%
BenchmarkEscapeHost/http_hyphen-8	142	168	+18.31%
BenchmarkEscapeHost/https_hyphen-8	133	170	+27.82%
BenchmarkEscapeHost/1hyphen-8	53.3	165	+209.57%
BenchmarkEscapeHost/2hyphen-8	56.8	163	+186.97%
BenchmarkEscapeHost/1dot-8	124	177	+42.74%
BenchmarkEscapeHost/2dot-8	156	200	+28.21%
BenchmarkEscapeHost/mixed-8	142	172	+21.13%
BenchmarkEscapeHost/full-8	219	265	+21.00%
BenchmarkEscapeHost/weird-8	351	263	-25.07%
BenchmarkEscapeHost/prefix-8	111	149	+34.23%
BenchmarkEscapeHost/suffix-8	100	146	+46.00%
BenchmarkEscapeHost/none-8	38.1	130	+241.21%

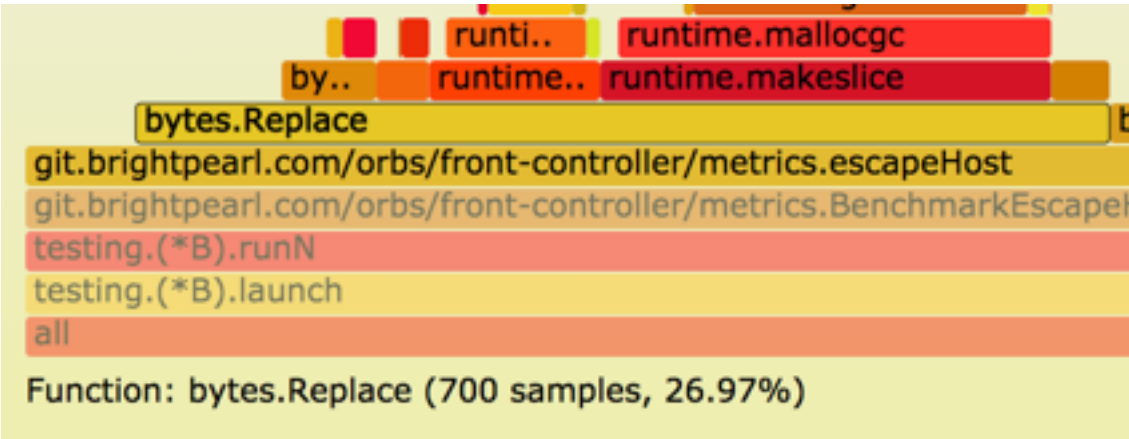
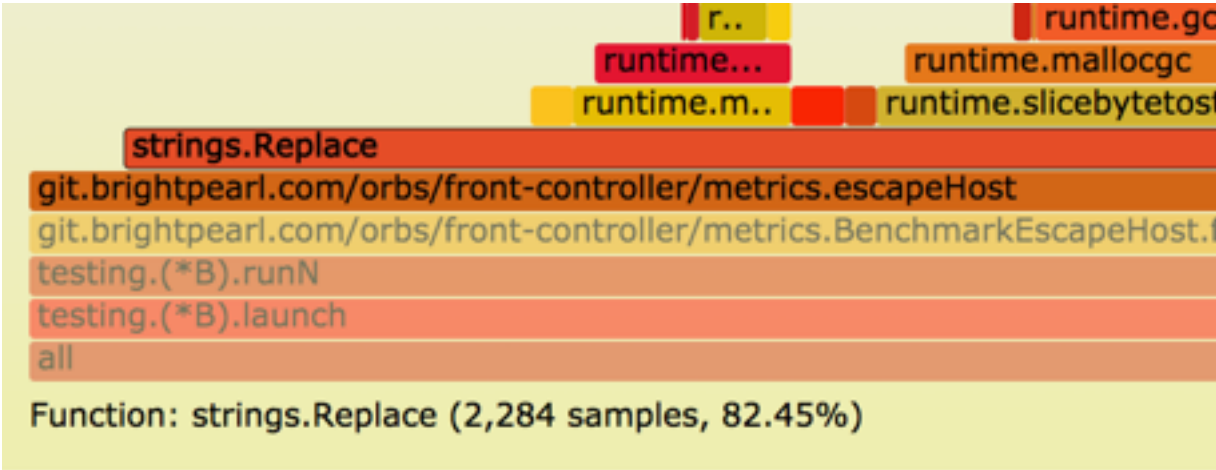
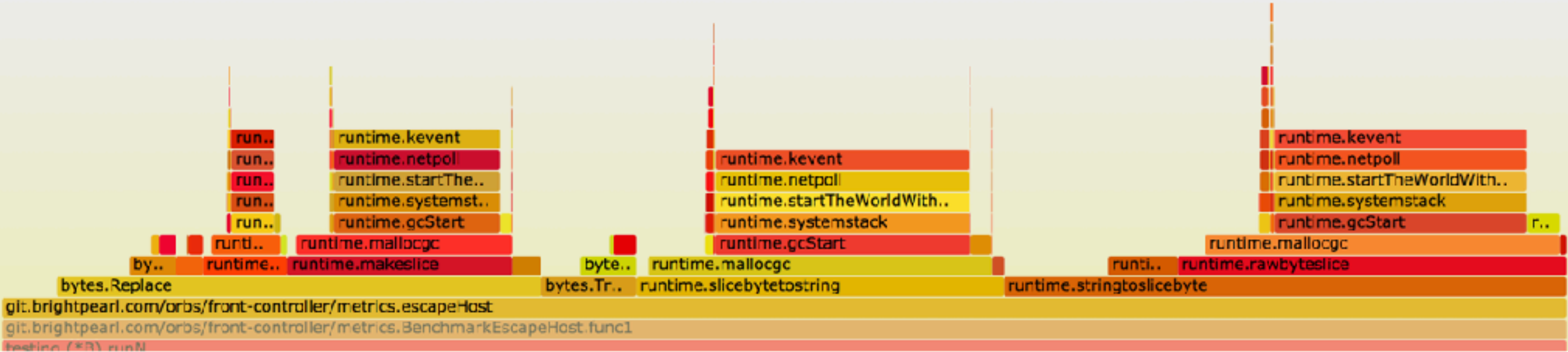
benchmark	old allocs	new allocs	delta
BenchmarkEscapeHost/empty-8	0	0	+0.00%
BenchmarkEscapeHost/http_dot-8	4	3	-25.00%
BenchmarkEscapeHost/https_dot-8	4	3	-25.00%
BenchmarkEscapeHost/http_hyphen-8	2	3	+50.00%
BenchmarkEscapeHost/https_hyphen-8	2	3	+50.00%
BenchmarkEscapeHost/1hyphen-8	0	3	+Inf%
BenchmarkEscapeHost/2hyphen-8	0	3	+Inf%
BenchmarkEscapeHost/1dot-8	2	3	+50.00%
BenchmarkEscapeHost/2dot-8	2	3	+50.00%
BenchmarkEscapeHost/mixed-8	2	3	+50.00%
BenchmarkEscapeHost/full-8	2	3	+50.00%
BenchmarkEscapeHost/weird-8	4	3	-25.00%
BenchmarkEscapeHost/prefix-8	2	3	+50.00%
BenchmarkEscapeHost/suffix-8	2	3	+50.00%
BenchmarkEscapeHost/none-8	0	3	+Inf%

benchmark	old bytes	new bytes	delta
BenchmarkEscapeHost/empty-8	0	0	+0.00%
BenchmarkEscapeHost/http_dot-8	96	80	-16.67%
BenchmarkEscapeHost/https_dot-8	96	80	-16.67%
BenchmarkEscapeHost/http_hyphen-8	32	80	+150.00%
BenchmarkEscapeHost/https_hyphen-8	32	80	+150.00%
BenchmarkEscapeHost/1hyphen-8	0	48	+Inf%
BenchmarkEscapeHost/2hyphen-8	0	48	+Inf%
BenchmarkEscapeHost/1dot-8	32	48	+50.00%
BenchmarkEscapeHost/2dot-8	32	48	+50.00%
BenchmarkEscapeHost/mixed-8	32	48	+50.00%
BenchmarkEscapeHost/full-8	128	192	+50.00%
BenchmarkEscapeHost/weird-8	256	192	-25.00%
BenchmarkEscapeHost/prefix-8	10	24	+140.00%
BenchmarkEscapeHost/suffix-8	10	24	+140.00%
BenchmarkEscapeHost/none-8	0	24	+Inf%

Reset Zoom

Flame Graph

Search




```
func escapeHost(host string) []byte {  
    escaped := bytes.Replace([]byte(host), []byte("."), []byte("-"), -1)  
    escaped = bytes.TrimPrefix(escaped, []byte("http://"))  
    escaped = bytes.TrimPrefix(escaped, []byte("https://"))  
  
    return escaped  
}
```

```
func (m *Handler) AddLocatorTime(host string, d time.Duration) {  
    metrics.GetOrRegisterTimer(  
        fmt.Sprintf("LocatorTime.%s", escapeHost(host)),  
        nil  
    ).Update(d)  
}
```

```
$ benchcmp baseline.txt return-byte.txt
```

benchmark	old ns/op	new ns/op	delta
BenchmarkEscapeHost/empty-8	34.9	63.9	+83.09%
BenchmarkEscapeHost/http_dot-8	212	152	-28.30%
BenchmarkEscapeHost/https_dot-8	233	152	-34.76%
BenchmarkEscapeHost/http_hyphen-8	142	134	-5.63%
BenchmarkEscapeHost/https_hyphen-8	133	135	+1.50%
BenchmarkEscapeHost/1hyphen-8	53.3	124	+132.65%
BenchmarkEscapeHost/2hyphen-8	56.8	126	+121.83%
BenchmarkEscapeHost/1dot-8	124	142	+14.52%
BenchmarkEscapeHost/2dot-8	156	159	+1.92%
BenchmarkEscapeHost/mixed-8	142	143	+0.70%
BenchmarkEscapeHost/full-8	219	221	+0.91%
BenchmarkEscapeHost/weird-8	351	222	-36.75%
BenchmarkEscapeHost/prefix-8	111	125	+12.61%
BenchmarkEscapeHost/suffix-8	100	137	+37.00%
BenchmarkEscapeHost/none-8	38.1	118	+209.71%

benchmark	old allocs	new allocs	delta
BenchmarkEscapeHost/empty-8	0	0	+0.00%
BenchmarkEscapeHost/http_dot-8	4	2	-50.00%
BenchmarkEscapeHost/https_dot-8	4	2	-50.00%
BenchmarkEscapeHost/http_hyphen-8	2	2	+0.00%
BenchmarkEscapeHost/https_hyphen-8	2	2	+0.00%
BenchmarkEscapeHost/1hyphen-8	0	2	+Inf%
BenchmarkEscapeHost/2hyphen-8	0	2	+Inf%
BenchmarkEscapeHost/1dot-8	2	2	+0.00%
BenchmarkEscapeHost/2dot-8	2	2	+0.00%
BenchmarkEscapeHost/mixed-8	2	2	+0.00%
BenchmarkEscapeHost/full-8	2	2	+0.00%
BenchmarkEscapeHost/weird-8	4	2	-50.00%
BenchmarkEscapeHost/prefix-8	2	2	+0.00%
BenchmarkEscapeHost/suffix-8	2	2	+0.00%
BenchmarkEscapeHost/none-8	0	2	+Inf%

benchmark	old bytes	new bytes	delta
BenchmarkEscapeHost/empty-8	0	0	+0.00%
BenchmarkEscapeHost/http_dot-8	96	64	-33.33%
BenchmarkEscapeHost/https_dot-8	96	64	-33.33%
BenchmarkEscapeHost/http_hyphen-8	32	64	+100.00%
BenchmarkEscapeHost/https_hyphen-8	32	64	+100.00%
BenchmarkEscapeHost/1hyphen-8	0	32	+Inf%
BenchmarkEscapeHost/2hyphen-8	0	32	+Inf%
BenchmarkEscapeHost/1dot-8	32	32	+0.00%
BenchmarkEscapeHost/2dot-8	32	32	+0.00%
BenchmarkEscapeHost/mixed-8	32	32	+0.00%
BenchmarkEscapeHost/full-8	128	128	+0.00%
BenchmarkEscapeHost/weird-8	256	128	-50.00%
BenchmarkEscapeHost/prefix-8	10	16	+60.00%
BenchmarkEscapeHost/suffix-8	10	16	+60.00%
BenchmarkEscapeHost/none-8	0	16	+Inf%

Reset Zoom

Flame Graph

Search



```
func escapeHost(host string) []byte {  
    escaped := bytes.Replace([]byte(host), []byte("."), []byte("-"), -1)  
    escaped = bytes.TrimPrefix(escaped, []byte("http://"))  
    return bytes.TrimPrefix(escaped, []byte("https://"))  
}
```

```

$ benchcmp return-byte.txt return-away.txt
benchmark                old ns/op    new ns/op    delta
BenchmarkEscapeHost/empty-8    63.9         63.8         -0.16%
BenchmarkEscapeHost/http_dot-8  152          151          -0.66%
BenchmarkEscapeHost/https_dot-8 152          152          +0.00%
BenchmarkEscapeHost/http_hyphen-8 134          134          +0.00%
BenchmarkEscapeHost/https_hyphen-8 135          134          -0.74%
BenchmarkEscapeHost/1hyphen-8   124          127          +2.42%
BenchmarkEscapeHost/2hyphen-8   126          127          +0.79%
BenchmarkEscapeHost/1dot-8      142          152          +7.04%
BenchmarkEscapeHost/2dot-8      159          176          +10.69%
BenchmarkEscapeHost/mixed-8     143          150          +4.90%
BenchmarkEscapeHost/full-8      221          232          +4.98%
BenchmarkEscapeHost/weird-8     222          230          +3.60%
BenchmarkEscapeHost/prefix-8    125          128          +2.40%
BenchmarkEscapeHost/suffix-8    137          129          -5.84%
BenchmarkEscapeHost/none-8      118          107          -9.32%

benchmark                old allocs   new allocs   delta
BenchmarkEscapeHost/empty-8    0            0            +0.00%
BenchmarkEscapeHost/http_dot-8  2            2            +0.00%
BenchmarkEscapeHost/https_dot-8  2            2            +0.00%
BenchmarkEscapeHost/http_hyphen-8  2            2            +0.00%
BenchmarkEscapeHost/https_hyphen-8  2            2            +0.00%
BenchmarkEscapeHost/1hyphen-8   2            2            +0.00%
BenchmarkEscapeHost/2hyphen-8   2            2            +0.00%
BenchmarkEscapeHost/1dot-8      2            2            +0.00%
BenchmarkEscapeHost/2dot-8      2            2            +0.00%
BenchmarkEscapeHost/mixed-8     2            2            +0.00%
BenchmarkEscapeHost/full-8      2            2            +0.00%
BenchmarkEscapeHost/weird-8     2            2            +0.00%
BenchmarkEscapeHost/prefix-8    2            2            +0.00%
BenchmarkEscapeHost/suffix-8    2            2            +0.00%
BenchmarkEscapeHost/none-8      2            2            +0.00%

benchmark                old bytes    new bytes    delta
BenchmarkEscapeHost/empty-8    0            0            +0.00%
BenchmarkEscapeHost/http_dot-8  64           64           +0.00%
BenchmarkEscapeHost/https_dot-8  64           64           +0.00%
BenchmarkEscapeHost/http_hyphen-8  64           64           +0.00%
BenchmarkEscapeHost/https_hyphen-8  64           64           +0.00%
BenchmarkEscapeHost/1hyphen-8   32           32           +0.00%
BenchmarkEscapeHost/2hyphen-8   32           32           +0.00%
BenchmarkEscapeHost/1dot-8      32           32           +0.00%
BenchmarkEscapeHost/2dot-8      32           32           +0.00%
BenchmarkEscapeHost/mixed-8     32           32           +0.00%
BenchmarkEscapeHost/full-8      128          128          +0.00%
BenchmarkEscapeHost/weird-8     128          128          +0.00%
BenchmarkEscapeHost/prefix-8    16           16           +0.00%
BenchmarkEscapeHost/suffix-8    16           16           +0.00%
BenchmarkEscapeHost/none-8      16           16           +0.00%

```

```
func escapeHost(host string) []byte {  
    escaped := bytes.Replace([]byte(host), []byte("."), []byte("-"), -1)  
    escaped = bytes.TrimPrefix(escaped, []byte("http://"))  
    return bytes.TrimPrefix(escaped, []byte("https://"))  
}
```

```
$ go build -gcflags='-m' main.go  
# command-line-arguments  
./main.go:13:33: ([]byte)(host) escapes to heap  
./main.go:12:32: escapeHost host does not escape  
./main.go:13:47: escapeHost ([]byte)(".") does not escape  
./main.go:13:60: escapeHost ([]byte)("-") does not escape  
./main.go:14:44: escapeHost ([]byte)("http://") does not escape  
./main.go:15:41: escapeHost ([]byte)("https://") does not escape  
./main.go:9:24: escapeHost("http://some.weird-host") escapes to  
heap  
./main.go:9:13: main ... argument does not escape
```

TAKEAWAYS

PRIORITISE

PRIORITISE

CUSTOMISE

PRIORITISE

CUSTOMISE

MEASURE

PRIORITISE

CUSTOMISE

MEASURE

ITERATE

DESIGN > CODE
CHANGES

THANK YOU!



@kasiazien