

My tools for Gophers

May 2016, Barcelona


@idanyliuk

Tools for Gophers



Ivan Daniluk

divan

 Typeform






 Barcelona, Spain

 Contributions

 Repositories

 Public activity

Popular repositories

	expvarmon TermUI based monitor for Go apps using expv...	748 ★
	depscheck Dependency checker for Golang (Go) packag...	320 ★
	gobenchui UI for overview of your Golang package bench...	319 ★
	gofresh Keep your Go package dependencies fresh. C...	72 ★
	gorilla-xmlrpc Gorilla XML RPC implementation (Golang/Go)	15 ★

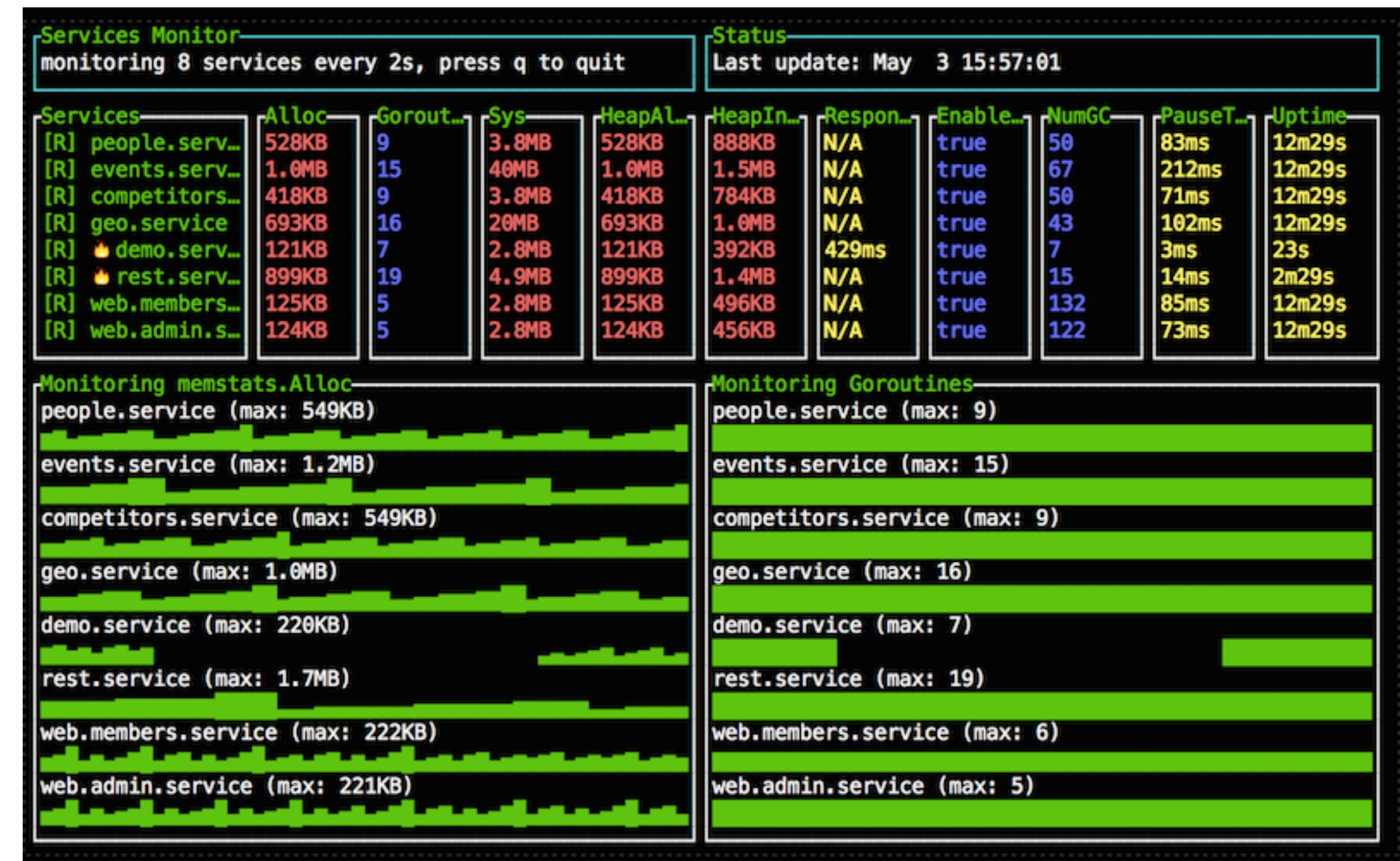
Expvarmon

zero-configuration monitoring for Go expvars

github.com/divan/expvarmon

Expvarmon

Expvar + monitoring



Expvar

- Stdlib package
- Easy to use
- Exposes **/debug/vars** endpoint

Package expvar

```
import "expvar"
```

[Overview](#)

[Index](#)

Overview ▼

Package expvar provides a standardized interface to public variables, such as operation counters in servers. It exposes these variables via HTTP at /debug/vars in JSON format.

Operations to set or modify these public variables are atomic.

In addition to adding the HTTP handler, this package registers the following variables:

```
cmdline    os.Args
memstats   runtime.Memstats
```

The package is sometimes only imported for the side effect of registering its HTTP handler and the above variables. To use it this way, link this package into your program:

```
import _ "expvar"
```

Easy to use

```
import _ "expvar"
```


Easy to use

```
import _ "expvar"

func goroutines() interface{} {
    return runtime.NumGoroutine()
}

expvar.Publish("Goroutines", expvar.Func(goroutines))
```


Expvarmon

- Zero configuration
- Polls /debug/vars endpoint
- Renders output as a charts in a console

Expvarmon

- No storage
- No search
- No notifications

Services Monitor

monitoring expvarmon every 200ms, press q to q...

Status

Last update: May 12 18:16:44

Alloc

1.8MB

Sys

8.9MB

HeapAlloc

1.8MB

HeapInuse

2.5MB

PauseNs

116μs

PauseTotalNs

2ms

Monitoring

memstats.Alloc: 1.8MB (max: 3.7MB)



memstats.Sys: 8.9MB (max: 8.9MB)



memstats.HeapAlloc: 1.8MB (max: 3.7MB)



memstats.HeapInuse: 2.5MB (max: 4.3MB)



memstats.PauseNs: 116μs (max: 280μs)



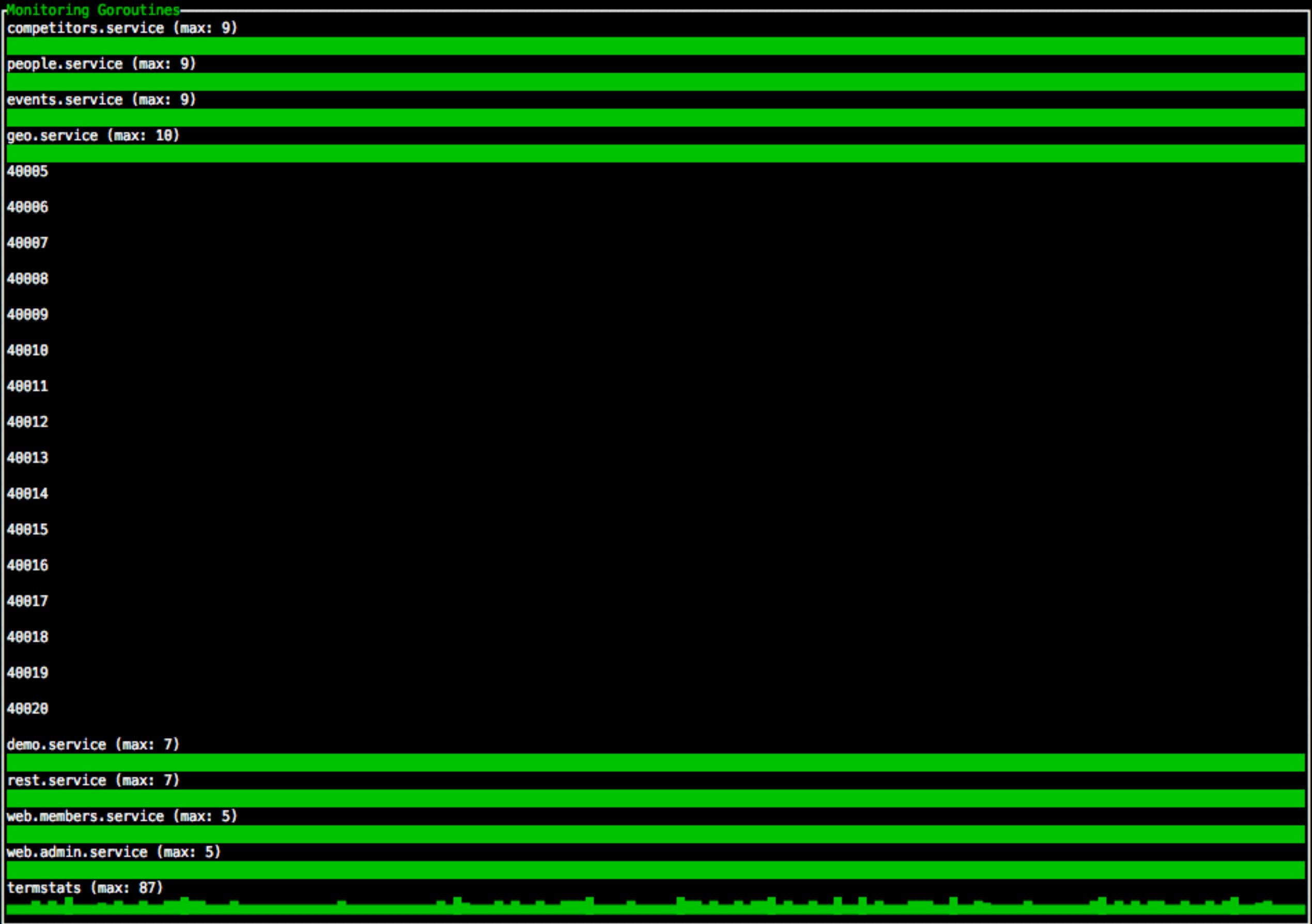
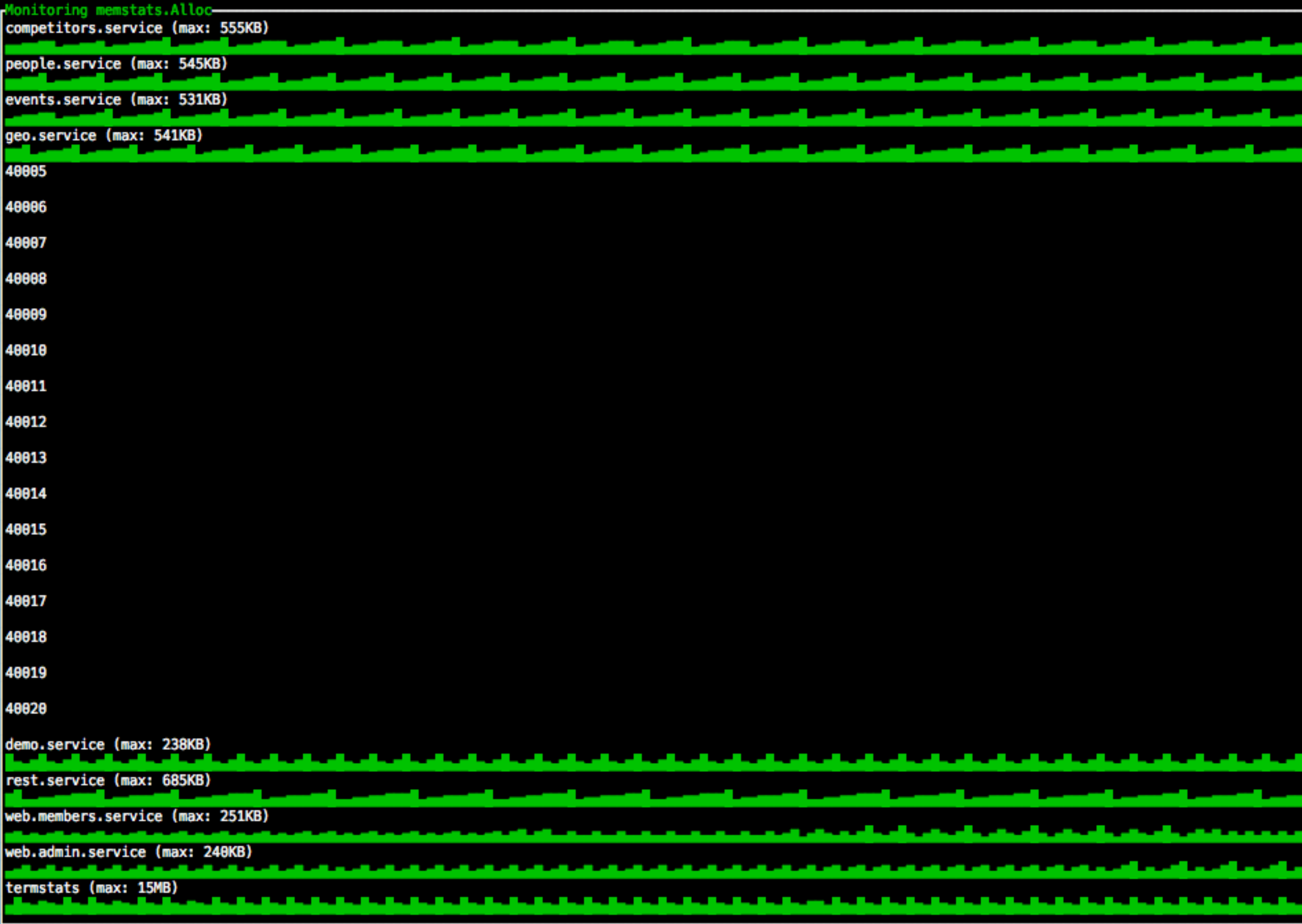
memstats.PauseTotalNs: 2ms (max: 2ms)



Services Monitor
monitoring 25 services every 200ms, press q to quit

Status
Last update: May 3 19:39:04

Services	Alloc	Goroutines	Sys	HeapAlloc	HeapInuse	ResponseTime	EnableGC	NumGC	PauseTotalNs	Uptime
[R] competitors.service	412KB	9	3.8MB	412KB	712KB	N/A	true	86	117ms	10m51s
[R] people.service	452KB	9	3.8MB	452KB	792KB	N/A	true	88	122ms	10m51s
[R] events.service	388KB	9	3.8MB	388KB	784KB	N/A	true	88	127ms	10m51s
[R] geo.service	500KB	10	3.8MB	500KB	824KB	N/A	true	93	114ms	10m51s
[E] 40005 failed	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
[E] 40006 failed	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
[E] 40007 failed	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
[E] 40008 failed	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
[E] 40009 failed	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
[E] 40010 failed	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
[E] 40011 failed	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
[E] 40012 failed	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
[E] 40013 failed	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
[E] 40014 failed	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
[E] 40015 failed	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
[E] 40016 failed	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
[E] 40017 failed	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
[E] 40018 failed	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
[E] 40019 failed	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
[E] 40020 failed	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
[R] demo.service	238KB	7	2.8MB	238KB	536KB	486ms	true	1189	788ms	3h42m26s
[R] rest.service	518KB	7	3.8MB	518KB	984KB	N/A	true	52	70ms	3m23s
[R] web.members.service	187KB	5	2.8MB	187KB	520KB	N/A	true	243	154ms	10m51s
[R] web.admin.service	174KB	5	2.8MB	174KB	496KB	N/A	true	210	123ms	10m51s
[R] termstats	10MB	51	20MB	10MB	11MB	N/A	true	131	745ms	36s



DEMO

Configuration

- Almost zero configuration, cmdline only
- Ports/URLs to monitor (*-ports*)
- JSON fields to monitor (*-vars*)
- Interval (*-i*)

Configuration - Ports

- 8080
- 8080-8082
- localhost:1234-1235,http://remote.app:1234
- http://user:auth@remote:1234-1238

Configuration - Vars

- Nested JSON fields are comma separated
- `-vars = "memstat.Alloc,memstat.PauseNS"`
- `-vars = "Goroutines,mycounter"`
- `-vars = "mem:memstat.HeapInuse,duration:memstat.PauseNS"`

Configuration - Modifiers

- mem: - treats as memory (renders as KB, MB, GB, etc.)
- duration: - treats as time duration (renders as μ s, ms, s, etc)
- string: - treats as string and don't display chart

Use cases

- debug session (see goroutines, mem, etc)
- quick look into running services

Security

- Don't expose '/debug/vars' to internet
- Use different http.Server or framework, if needed
- Consider sane -i option to not overload the service

DepsCheck

LeftPad eliminator

github.com/divan/depscheck

```
2. bash
[mac@~]$ depscheck -v github.com/divan/depscheck
github.com/divan/depscheck: 2 packages, 1041 LOC, 25 calls, 0 depth, 35 depth int.
```

PKG	RECV	NAME	TYPE	COUNT	LOC	LOCCUM	DEPTH	DEPTHINT
loader	*Config	FromArgs	method	1	32	52	0	5
	*Config	Load	method	1	190	571	0	20
	*Program	InitialPackages	method	3	7	7	0	0
	*Program	Package	method	2	10	10	0	0
		FromArgsUsage	const	1				
		Config	type	1				
		PackageInfo	type	6				
tablewriter		Program	type	2				
	*Table	Append	method	2	19	67	0	1
	*Table	SetHeader	method	2	6	54	0	1
	Table	Render	method	2	12	257	0	8
		NewWriter	func	2	23	23	0	0

PKG	PATH	COUNT	CALLS	LOCCUM	DEPTH	DEPTHINT
loader	golang.org/x/tools/go/loader	8	17	640	0	25
tablewriter	github.com/olekukonko/tablewriter	4	8	401	0	10

```
Cool, looks like your dependencies are sane.
[mac@~]$
```

LeftPad and Go

- Blog post - [LeftPad and Go: can tooling help?](#)
- Inspired by LeftPad fiasco story in JS community.
- Helps to find small dependencies

Dependencies

- Pros: eliminate complexity (short-term)
- Cons: add complexity (long-term)
- DRY principle is about abstractions, not about code
- Key is to manage and control complexity of the project

DepsCheck

- Why wait for the experience to come?
- Let's allow tools to help us.
- DepsCheck analyzes AST of your program and its deps
- And shows candidates for 'small dependencies'


```
[mac@~]$ depscheck -v github.com/divan/depscheck
```

```
github.com/divan/depscheck: 2 packages, 1041 LOC, 25 calls, 0 depth, 35 depth int.
```

PKG	RECV	NAME	TYPE	COUNT	LOC	LOCCUM	DEPTH	DEPTHINT
loader	*Config	FromArgs	method	1	32	52	0	5
	*Config	Load	method	1	190	571	0	20
	*Program	InitialPackages	method	3	7	7	0	0
	*Program	Package	method	2	10	10	0	0
		FromArgsUsage	const	1				
		Config	type	1				
		PackageInfo	type	6				
		Program	type	2				
tablewriter	*Table	Append	method	2	19	67	0	1
	*Table	SetHeader	method	2	6	54	0	1
	Table	Render	method	2	12	257	0	8
		NewWriter	func	2	23	23	0	0

PKG	PATH	COUNT	CALLS	LOCCUM	DEPTH	DEPTHINT
loader	golang.org/x/tools/go/loader	8	17	640	0	25
tablewriter	github.com/olekukonko/tablewriter	4	8	401	0	10

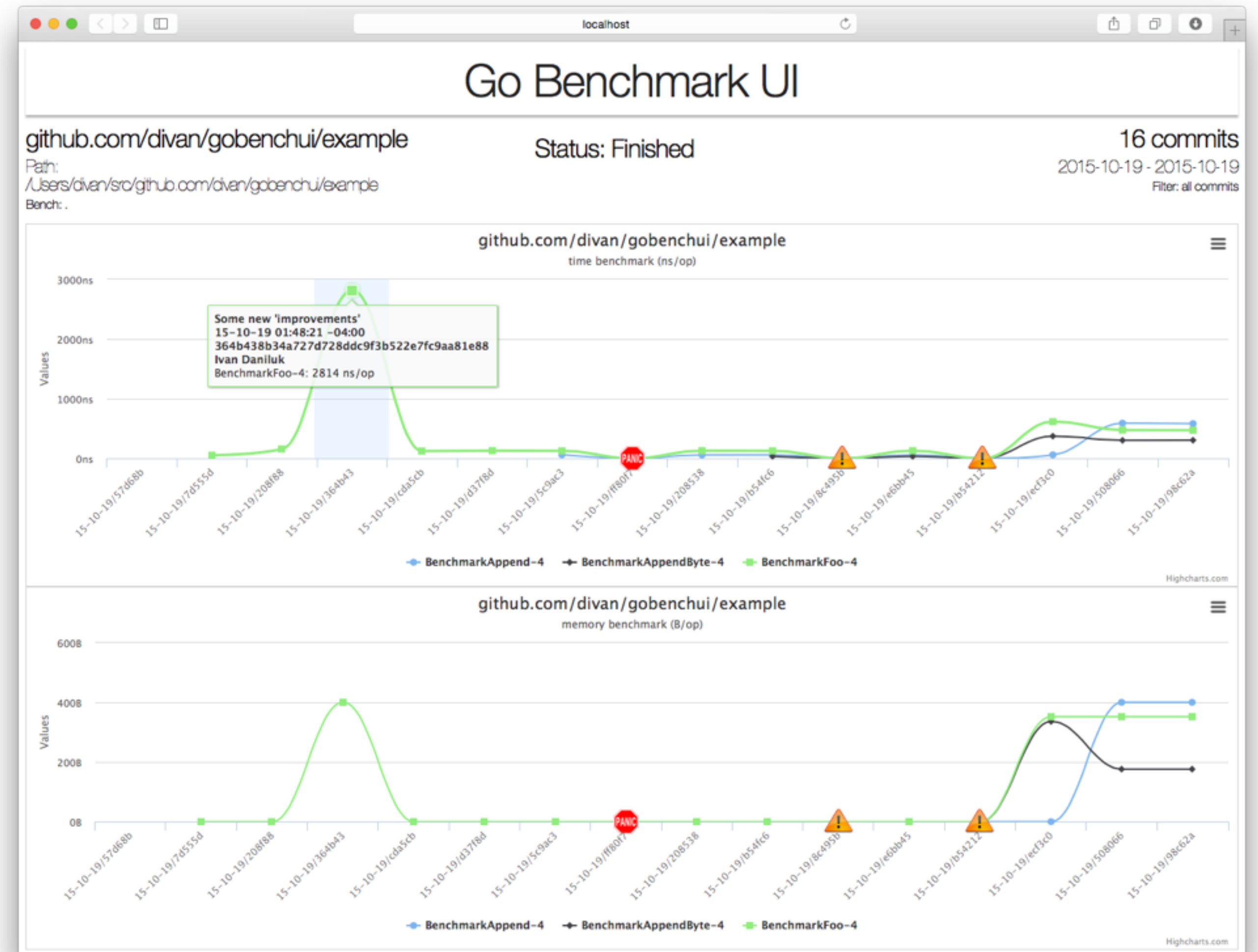
```
Cool, looks like your dependencies are sane.
```

```
[mac@~]$
```

DEMO

GoBenchUI

UI for benchmarks over time
github.com/divan/gobenchui



GoBenchUI

- UI for overview your package benchmarks progress
- Run benchmarks against every commit
- Display results using HighCharts.js graph

DEMO

GoFresh

Keep dependencies fresh
github.com/divan/gofresh

```
2. bash
[divan-mac.local@rest_api (M)]$ gofresh
Found 28 imports, checking for updates...
github.com/gocraft/web [✓50]
    0685b32 DOCUMENTATION: update readme to mention gocraft
    24976c2 Merge pull request #33 from DanielHeath/master
    31e2286 Support http.FileSystem as well as string as static middleware params
and 47 more...
git.apache.org/thrift.git/lib/go/thrift [✓550]
    113b638 THRIFT-3148 Markdown links to coding_standards are dead
    eab9c45 THRIFT-3089 Assigning default ENUM values results in non-compilable java code if j
ava namespace is not defined Client: Java Patch: Anna Dymek <aadymek@gmail.com>
    b9f6038 THRIFT-3146 Graphviz generates function name collisions between services Client: G
raphviz Patch: Adam Beberg
and 547 more...
github.com/go-sql-driver/mysql [✓71]
    0cc29e9 Merge pull request #332 from arnehormann/uint64params
    b2cd472 fix review issues, replace call with known value
    654304f Merge branch 'master' of https://github.com/go-sql-driver/mysql into colconv
and 68 more...

You have 3 packages out of date
To update all packages automatically, run gofresh -update
[divan-mac.local@rest_api (M)]$
```

GoFresh

- Checks your package imports for new updates
- Displays them to console
- Shows commits (aka Changelog)
- Easy wrapper for updating deps

[divan-mac.local@rest_api (M)]\$ gofresh

Found 28 imports, checking for updates...

github.com/gocraft/web [✓50]

0685b32 DOCUMENTATION: update readme to mention gocraft

24976c2 Merge pull request #33 from DanielHeath/master

31e2286 Support http.FileSystem as well as string as static middleware params

and 47 more...

git.apache.org/thrift.git/lib/go/thrift [✓550]

113b638 THRIFT-3148 Markdown links to coding_standards are dead

eab9c45 THRIFT-3089 Assigning default ENUM values results in non-compilable java code if j
ava namespace is not defined Client: Java Patch: Anna Dymek <aadymek@gmail.com>

b9f6038 THRIFT-3146 Graphviz generates function name collisions between services Client: G
raphviz Patch: Adam Beberg

and 547 more...

github.com/go-sql-driver/mysql [✓71]

0cc29e9 Merge pull request #332 from arnehormann/uint64params

b2cd472 fix review issues, replace call with known value

654304f Merge branch 'master' of https://github.com/go-sql-driver/mysql into colconv

and 68 more...

—
You have 3 packages out of date

To update all packages automatically, run gofresh -update

[divan-mac.local@rest_api (M)]\$

Workflow

- `cd src/github.com/user/project/`
- `$ gofresh`
- `$ gofresh -expand github.com/howeyc/fsnotify`
- `$ gofresh -update`
- `$ go test`

DEMO

Note

- Works only with deps in GOPATH or those ones that have VCS metadata kept

Thank you!