ECUG Con 十周年盛会

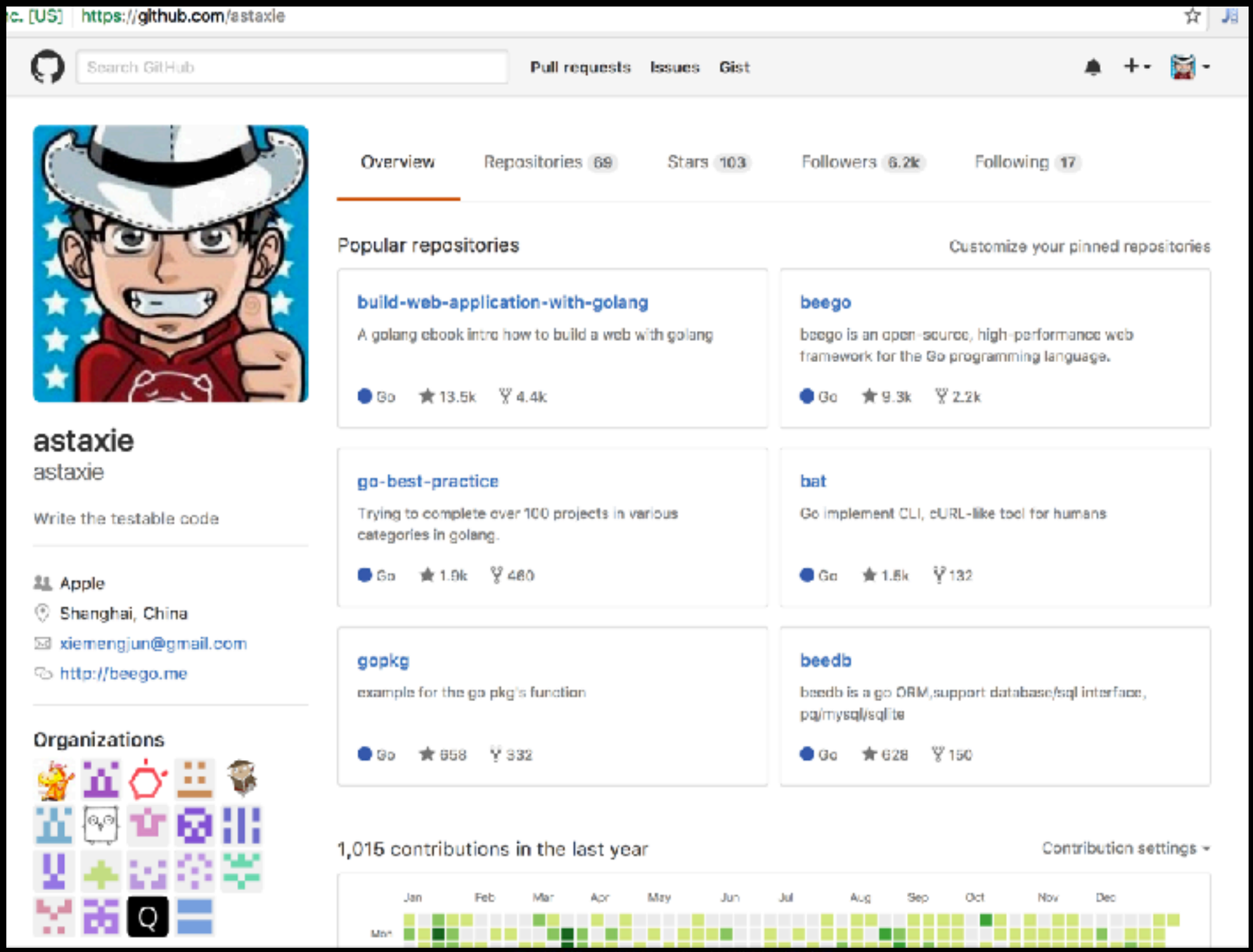
The State of Go

Asta

关于我

- ECUG 老顾客
- Gopher
- 开源爱好者

关于我


https://github.com/astaxie
Search GitHub
Pull requests
Issues
Gist
Overview
Repositories 69
Stars 103
Followers 6.2k
Following 17
Popular repositories
Customize your pinned repositories
build-web-application-with-golang
A golang ebook intro how to build a web with golang
Go 13.5k 4.4k
beego
beego is an open-source, high-performance web framework for the Go programming language.
Go 9.3k 2.2k
go-best-practice
Trying to complete over 100 projects in various categories in golang.
Go 1.9k 460
bat
Go implement CLI, cURL-like tool for humans
Go 1.8k 132
gopkg
example for the go pkg's function
Go 858 332
beedb
beedb is a go ORM,support database/sql interface, pq/mysql/sqlite
Go 628 150
astaxie
astaxie
Write the testable code
Apple
Shanghai, China
xiemengjun@gmail.com
http://beego.me
Organizations
1,015 contributions in the last year
Contribution settings
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Mon

Go回顾

- 2012/03 Go1 发布
- 2013/05 Go 1.1 发布
- 2013/12 Go 1.2 发布
- 2014/06 Go 1.3 发布
- 2014/12 Go 1.4 发布
- 2015/08 Go 1.5 发布
- 2016/02 Go 1.6 发布
- 2016/08 Go 1.7 发布

What's coming Go 1.8

- the language
- the standard library
- the runtime
- the tooling
- the community

Changes to the language

No changes on spec

```
type T1 struct {
    X int `json:"foo"`
}
type T2 struct {
    X int `json:"bar"`
}
var v1 T1
var v2 T2

v1 = T1(v2) // now legal
```

Changes to the tools

Compiler Toolchain

- 20-30% on 32-bit ARM systems
- Go 1.8 is about 15% faster.

go vet

stricter in some ways

copylocks for len&cap

```
func LenAndCapOnLockArrays() {
        var a [5]sync.Mutex
        aLen := len(a) // OK
        aCap := cap(a) // OK

        // override 'len' and 'cap' keywords

        len := func(interface{}) {}
        len(a) // ERROR "function call copies lock value: sync.Mutex"

        cap := func(interface{}) {}
        cap(a) // ERROR "function call copies lock value: sync.Mutex"
}
```

JSON tags

```
type Test struct {
        A string `json:"a"`
        B string `json:"a"`
        C string `json:"c",orm:"c"`
}
```

Close before checking errors

```
res, err := http.Get(“http://beego.me/robots.txt")
defer res.Body.Close()
if err != nil {
      log.Fatal(err)
}
```

Default GOPATH

- $HOME/go on Unix
- %USERPROFILE%/go on Windows

plugins

- loading plugins at runtime

Linux only

plugins

```
package main

func Add(x, y int) int {
    return x+y
}
```

```
go build -buildmode=plugin
```

```
go build -buildmode=plugin -o myplugin.so myplugin.go
```

plugins

```
p, _ := plugin.Open("./myplugin.so")
add, _ := p.Lookup("Add")

sum := add.(func(int, int) int)(1, 2)
```

new command

- go bug

go pprof

- suport TLS

Changes to runtime

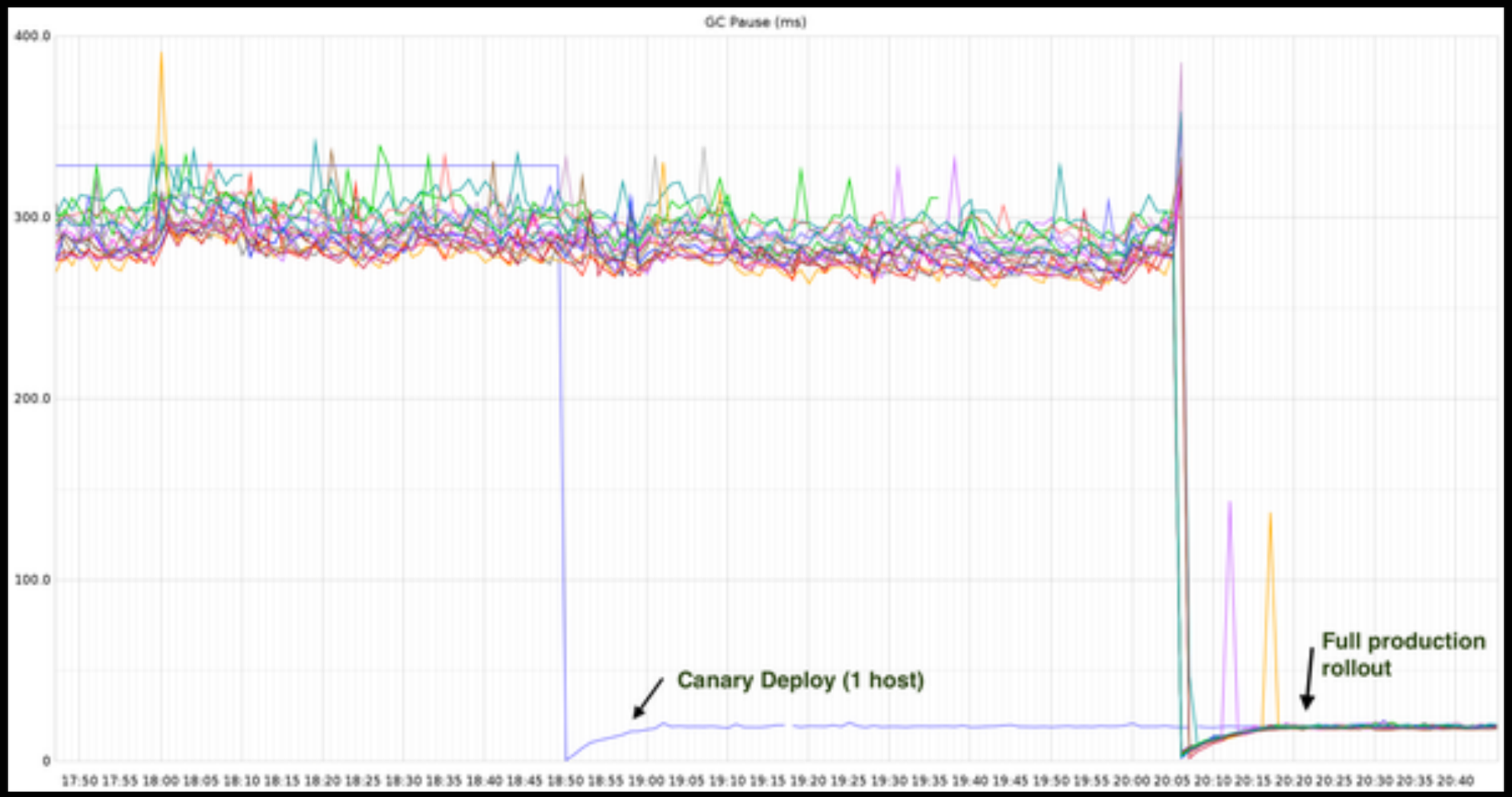
- Argument Liveness
- Concurrent Map Misuse
- MemStats Documentation

Performance

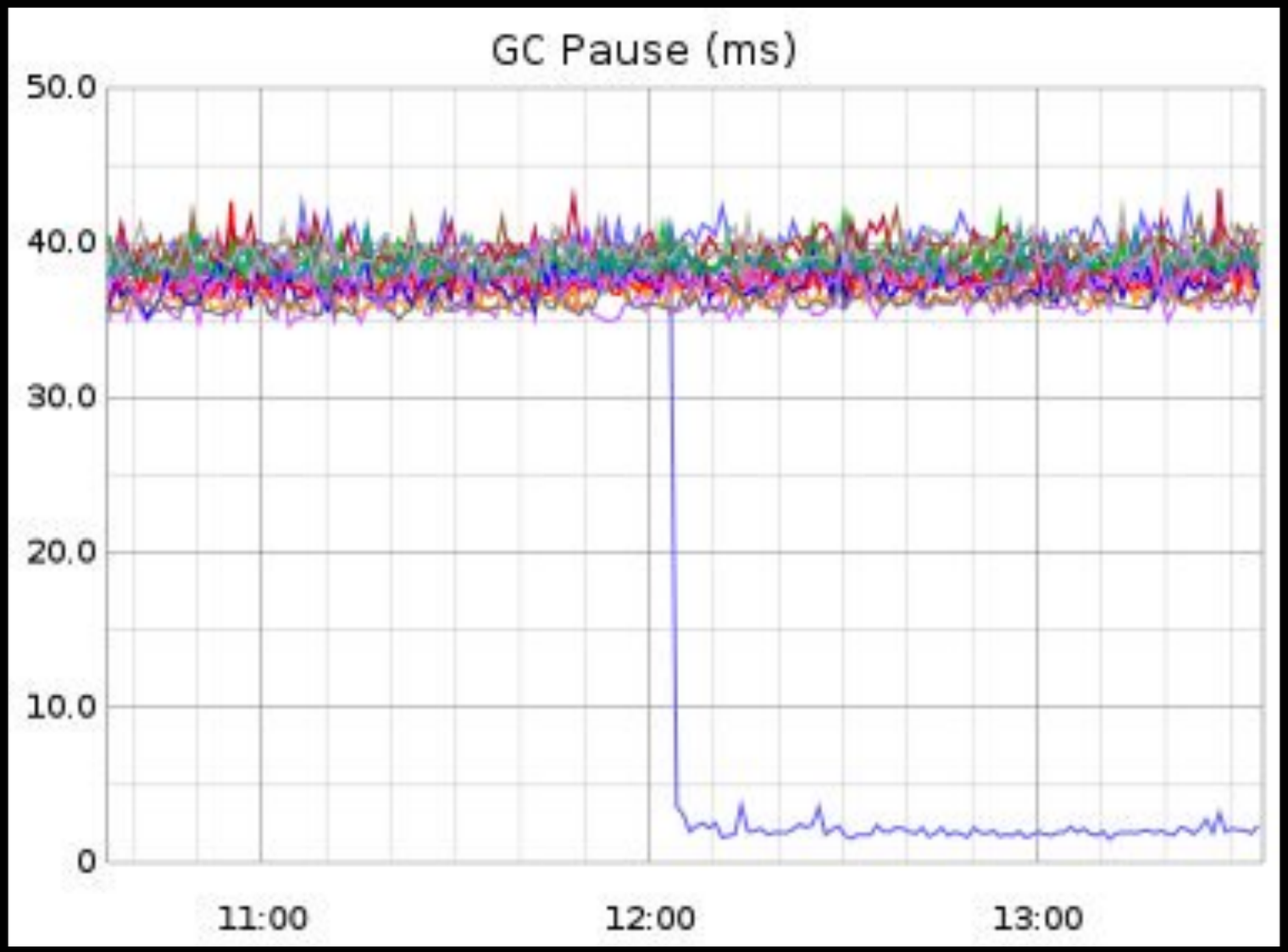
Standard library performance

- bytes,
- crypto/aes, crypto/cipher, crypto/elliptic, crypto/sha256, crypto/sha512,
- encoding/asn1, encoding/csv, encoding/hex, encoding/json,
- hash/crc32,
- image/color, image/draw,
- math, math/big, reflect, regexp, runtime,
- strconv, strings, syscall, text/template, and unicode/utf8

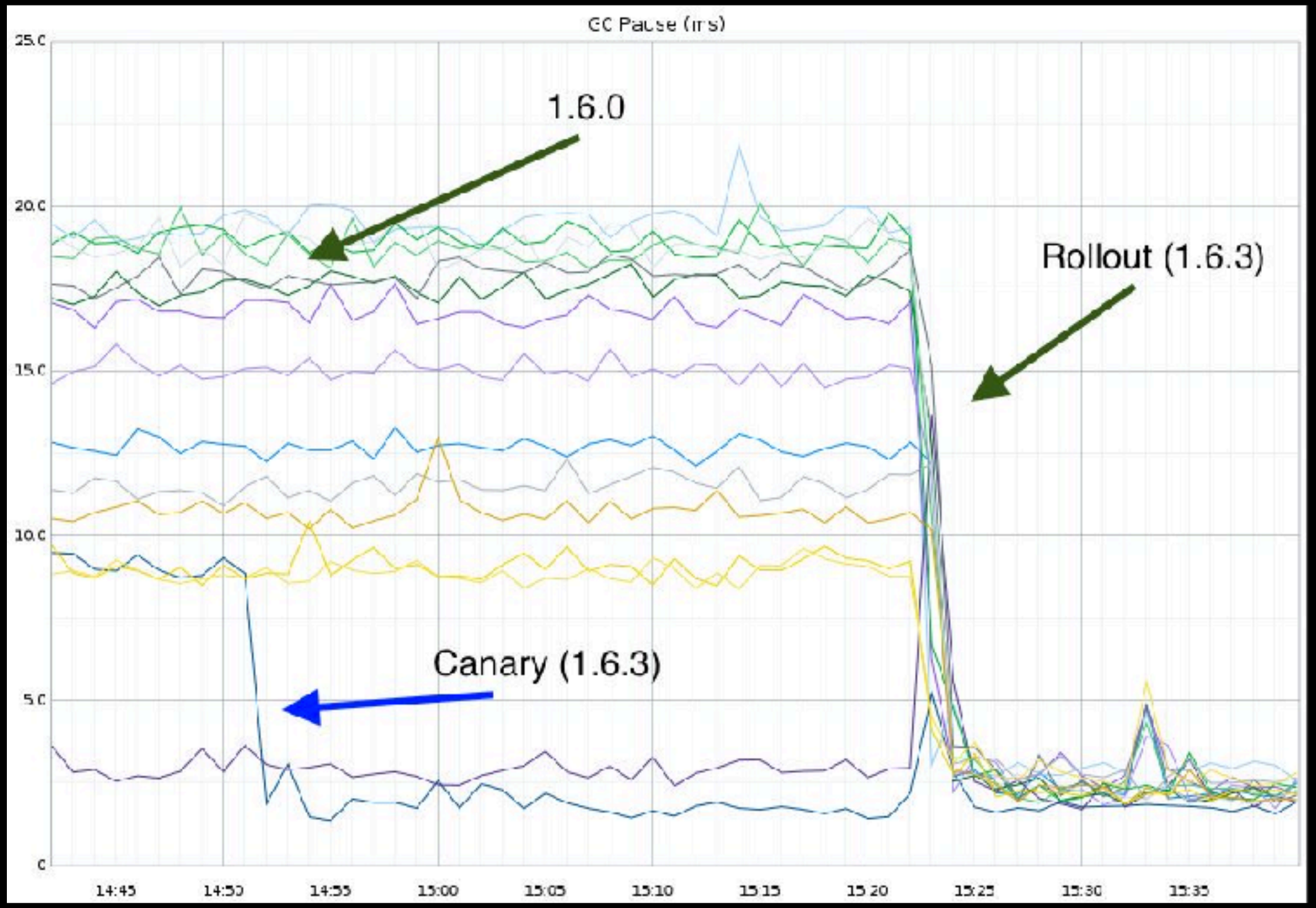
Garbage Collector



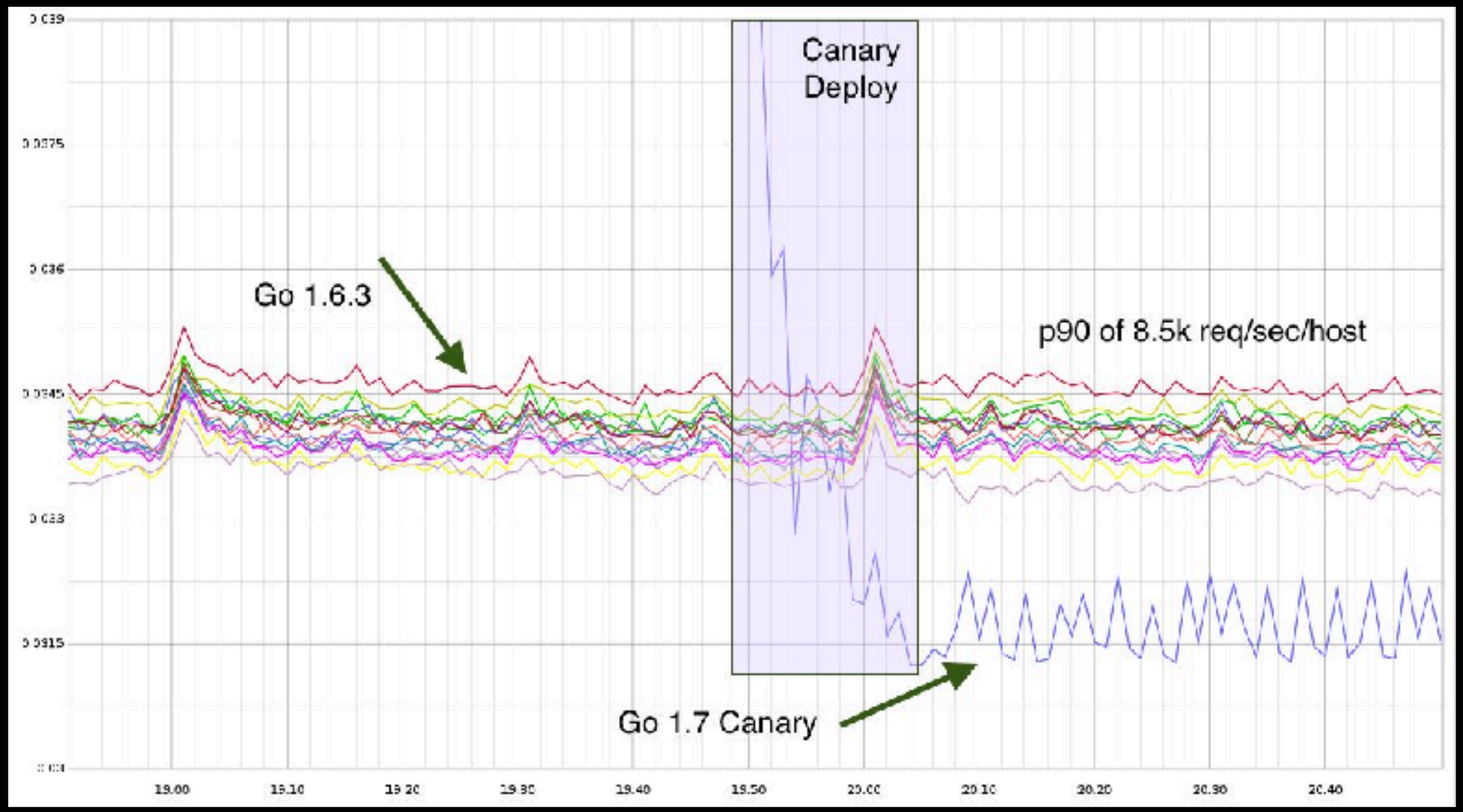
Garbage Collector



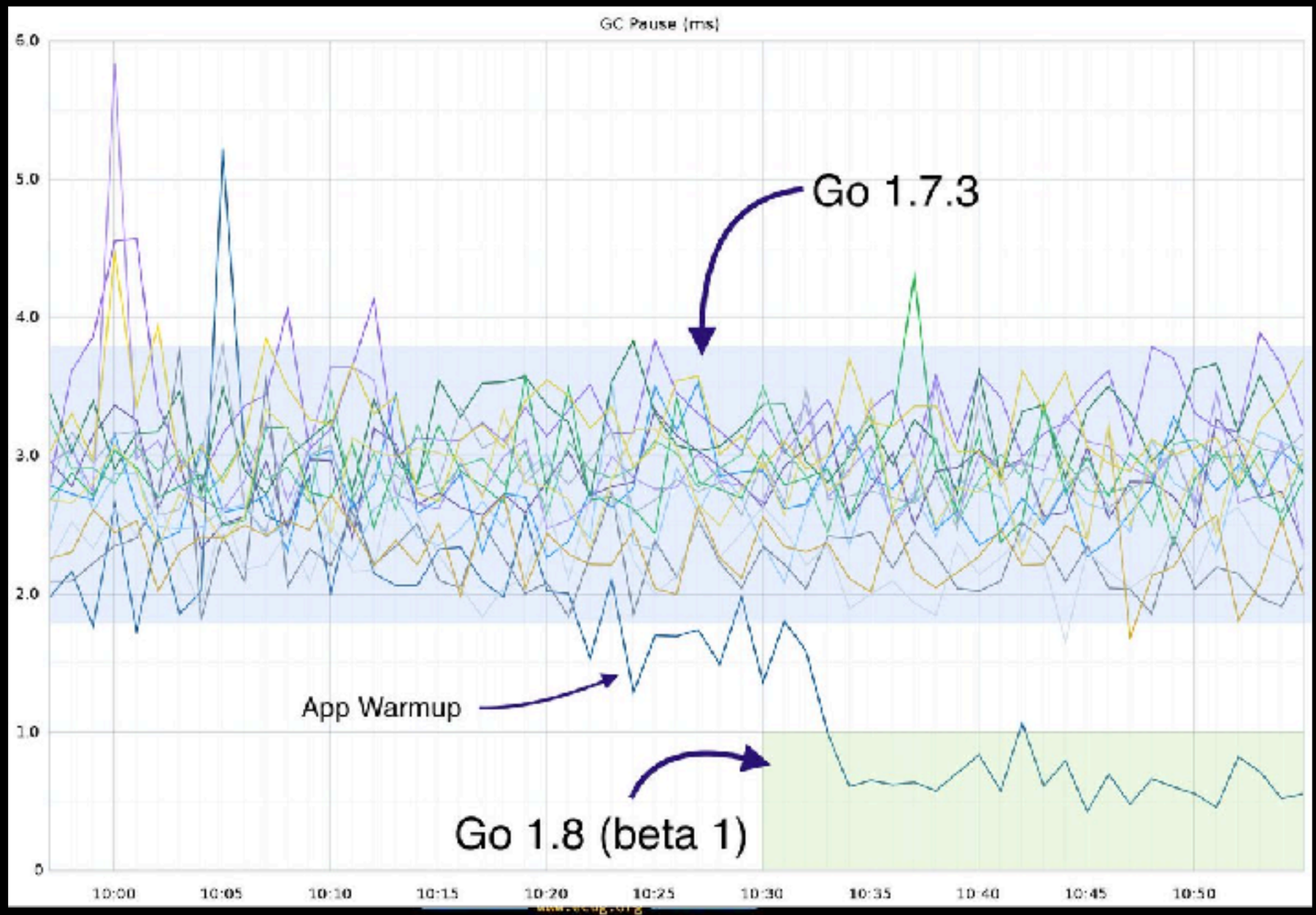
Garbage Collector



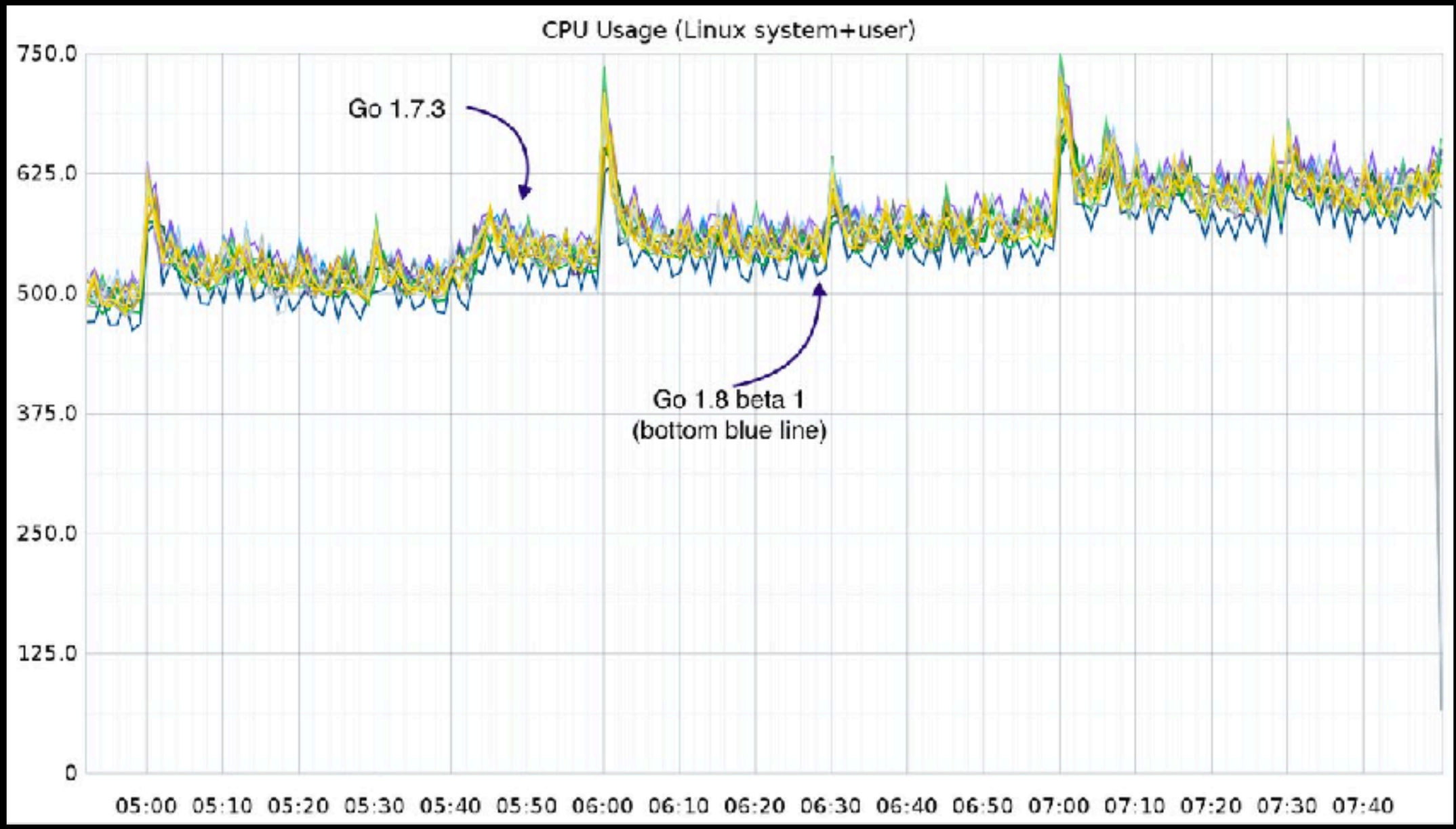
Garbage Collector



Garbage Collector



Garbage Collector



Defer

The overhead of deferred function calls has been reduced by about half.

CGO

The overhead of calls from Go into C has been reduced by about half.

New features

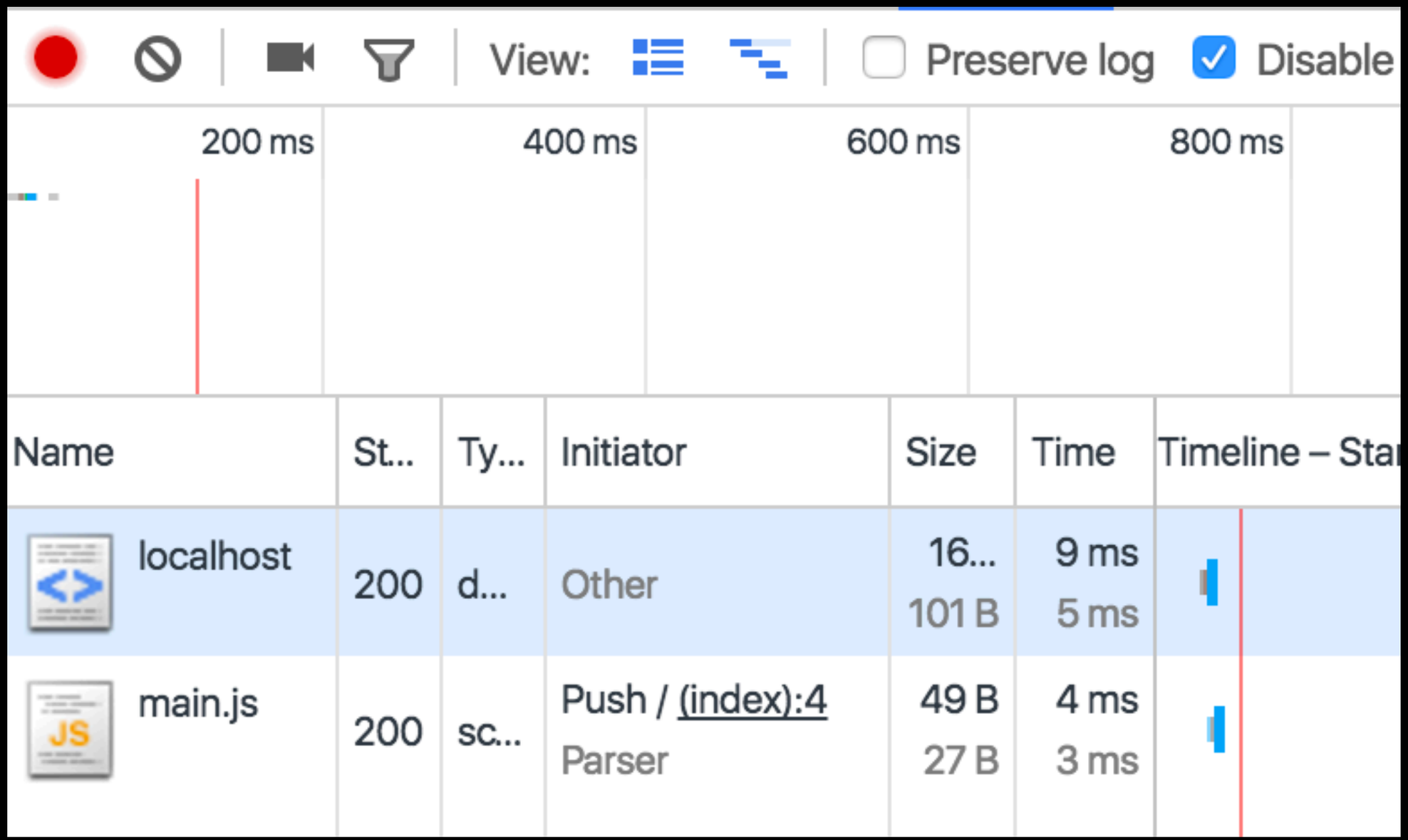
HTTP/2 Push

```
const indexHTML = `<html>
<head>
    <title>Hello</title>
    <script src="/main.js"></script>
</head>
<body>
</body>
</html>
`
```

HTTP/2 Push

```
http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
    if r.URL.Path != "/" {
        http.NotFound(w, r)
        return
    }
    pusher, ok := w.(http.Pusher)
    if ok { // Push is supported. Try pushing rather than waiting for the browser.
        if err := pusher.Push("/main.js", nil); err != nil {
            log.Printf("Failed to push: %v", err)
        }
    }
    fmt.Fprintf(w, indexHTML)
})
```

HTTP/2 Push


View:
Preserve log
Disable
200 ms
400 ms
600 ms
800 ms
Name
St...
Ty...
Initiator
Size
Time
Timeline – Sta
localhost
200
d...
Other
16...
101 B
9 ms
5 ms
main.js
200
sc...
Push / (index):4
Parser
49 B
27 B
4 ms
3 ms

Graceful Shutdown

```
// subscribe to SIGINT signals
stopChan := make(chan os.Signal)
signal.Notify(stopChan, os.Interrupt)

mux := http.NewServeMux()
mux.Handle("/", http.HandlerFunc(func(w http.ResponseWriter, r *http.Request) {
      time.Sleep(5 * time.Second)
      io.WriteString(w, "Finished!")
}))

srv := &http.Server{Addr: ":8081", Handler: mux}
```

Graceful Shutdown

```
go func() {
    // service connections
    if err := srv.ListenAndServe(); err != nil {
        log.Printf("listen: %s\n", err)
    }
}()

<-stopChan // wait for SIGINT
log.Println("Shutting down server...")

// shut down gracefully, but wait no longer than 5 seconds before halting
ctx, _ := context.WithTimeout(context.Background(), 5*time.Second)
srv.Shutdown(ctx)

log.Println("Server gracefully stopped")
```

Mutex Contention Profiling

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

var mu sync.Mutex
var items = make(map[int]struct{})

runtime.SetMutexProfileFraction(5)
for i := 0; i < 1000*1000; i++ {
    go func(i int) {
        mu.Lock()
        defer mu.Unlock()
        items[i] = struct{}{}
    }(i)
}

http.ListenAndServe(":8888", nil)
```

go tool pprof <binary> http://localhost:8888/debug/pprof/mutex?debug=1

```
Fetching profile from http://localhost:8888/debug/pprof/mutex
Saved profile in /Users/jbd/pprof/pprof.mutexprofile.localhost:8888.contentions.delay.002.pb
Entering interactive mode (type "help" for commands)
(pprof) list
Total: 27.15s
ROUTINE ======================== main.main.func1 in /Users/jbd/src/hello/mutexprofile/main.g
         0     27.15s (flat, cum)   100% of Total
         .          .     18:       go func() {
         .          .     19:           mu.Lock()
         .          .     20:           defer mu.Unlock()
         .          .     21:
         .          .     22:           items[i] = struct{}{}
         .     27.15s     23:       }()
         .          .     24:   }
         .          .     25:   http.ListenAndServe(":8888", nil)
         .          .     26:}
ROUTINE ======================== runtime.goexit in /Users/jbd/go/src/runtime/asm_amd64.s
         0     27.15s (flat, cum)   100% of Total
         .          .   2179:   RET
         .          .   2180:
         .          .   2181:// The top-most function running on a goroutine
         .          .   2182:// returns to goexit+PCQuantum.
         .          .   2183:TEXT runtime·goexit(SB),NOSPLIT,$0-0
         .     27.15s   2184:   BYTE    $0x90   // NOP
         .          .   2185:   CALL    runtime·goexit1(SB) // does not return
         .          .   2186:   // traceback from goexit1 must hit code range of goexit
         .          .   2187:   BYTE    $0x90   // NOP
         .          .   2188:
         .          .   2189:TEXT runtime·prefetcht0(SB),NOSPLIT,$0-8
ROUTINE ======================== sync.(*Mutex).Unlock in /Users/jbd/go/src/sync/mutex.go
    27.15s     27.15s (flat, cum)   100% of Total
         .          .    121:           return
         .          .    122:       }
         .          .    123:       // Grab the right to wake someone.
         .          .    124:       new = (old - 1<<mutexWaiterShift) | mutexWoken
         .          .    125:       if atomic.CompareAndSwapInt32(&m.state, old, new) {
    27.15s     27.15s    126:           runtime_Semrelease(&m.sema)
         .          .    127:           return
         .          .    128:       }
         .          .    129:       old = m.state
         .          .    130:   }
         .          .    131:}
```

database/sql

- Queries take a context.Context that can be used to cancel queries
- Native database column types are exposed via sql.ColumnType.
- Queries can be executed with named parameters if the underlying driver implements support for them.

New slice sorting API

```
type Peak struct {
	Name      string
	Elevation int // in feet
}

peaks := []Peak{
	{"Aconcagua", 22838},
	{"Denali", 20322},
	{"Kilimanjaro", 19341},
	{"Mount Elbrus", 18510},
	{"Mount Everest", 29029},
	{"Mount Kosciuszko", 7310},
	{"Mount Vinson", 16050},
	{"Puncak Jaya", 16024},
}

// does an in-place sort on the peaks slice, with tallest peak first
sort.Slice(peaks, func(i, j int) bool {
	return peaks[i].Elevation >= peaks[j].Elevation
})
```

Thanks


Gopher
认证