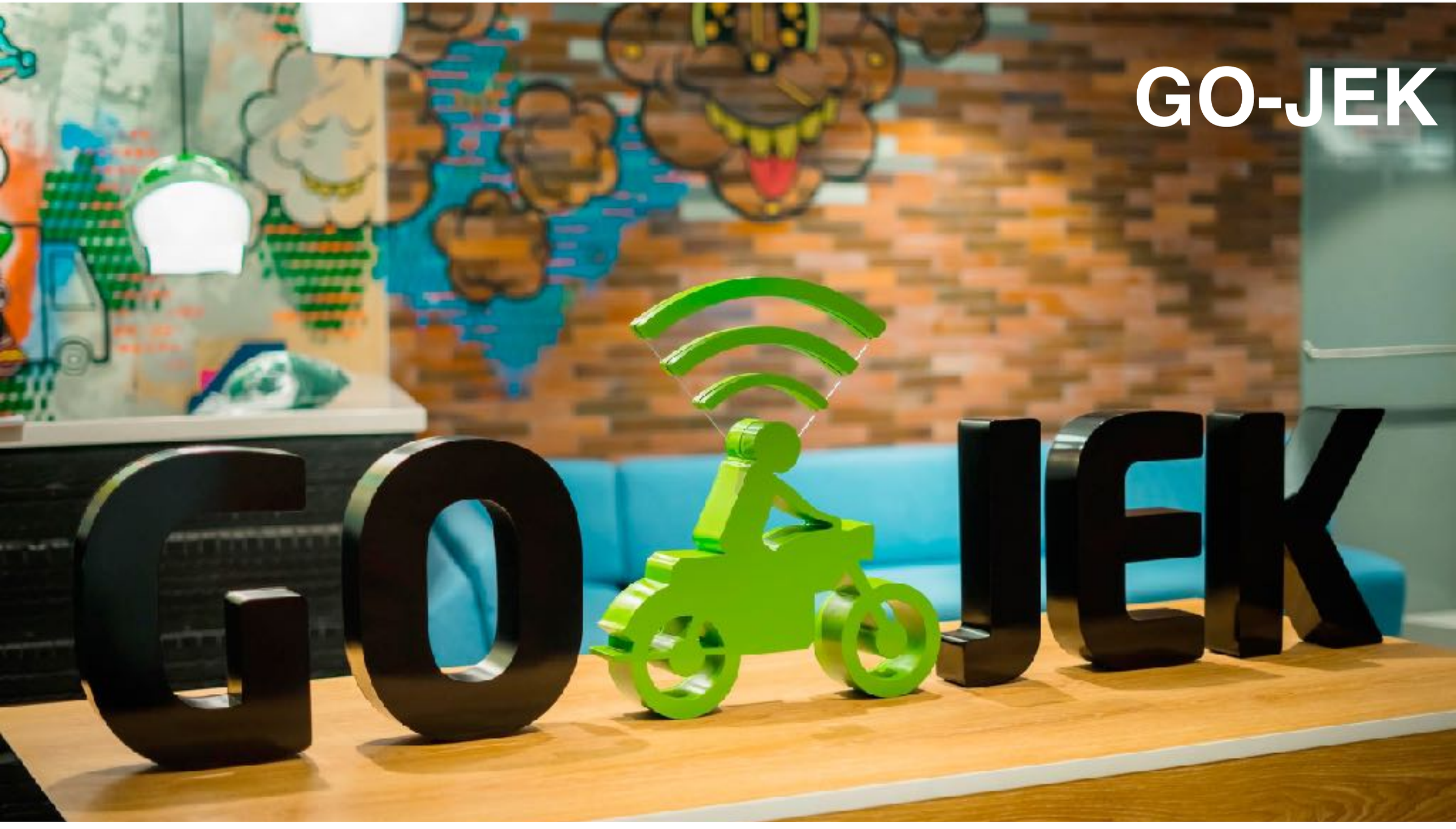

GO-JEK
GO JEK

Metric Tracking and Nuances of Memory Management

A bit more about me
Gopher @ Gojek
Love to code
Also known as Blackvirus

So what are we trying to do

GoFood

Stan Marsh

Legacy Monolith

↓

Performant Microservices

What is an order?

Interaction between 3 parties
Merchant, Driver, Customer

Created -> Waiting for Driver Allocation -> Driver OTW to pickup -> Driver Otw to Destination -> Complete

1. Create
2. GetState
3. Allocated
4. GetState
5. Pickup Food
6. Get State
7. Going to Destination
8. Get State
9. Complete
10. Get State


KILL
STAN MARSH

Safety Net

Metrics Tracking NewRelic Profiling

Safety Net

Metrics Tracking NewRelic Profiling

Safety Net

Metrics Tracking

NewRelic

Profiling

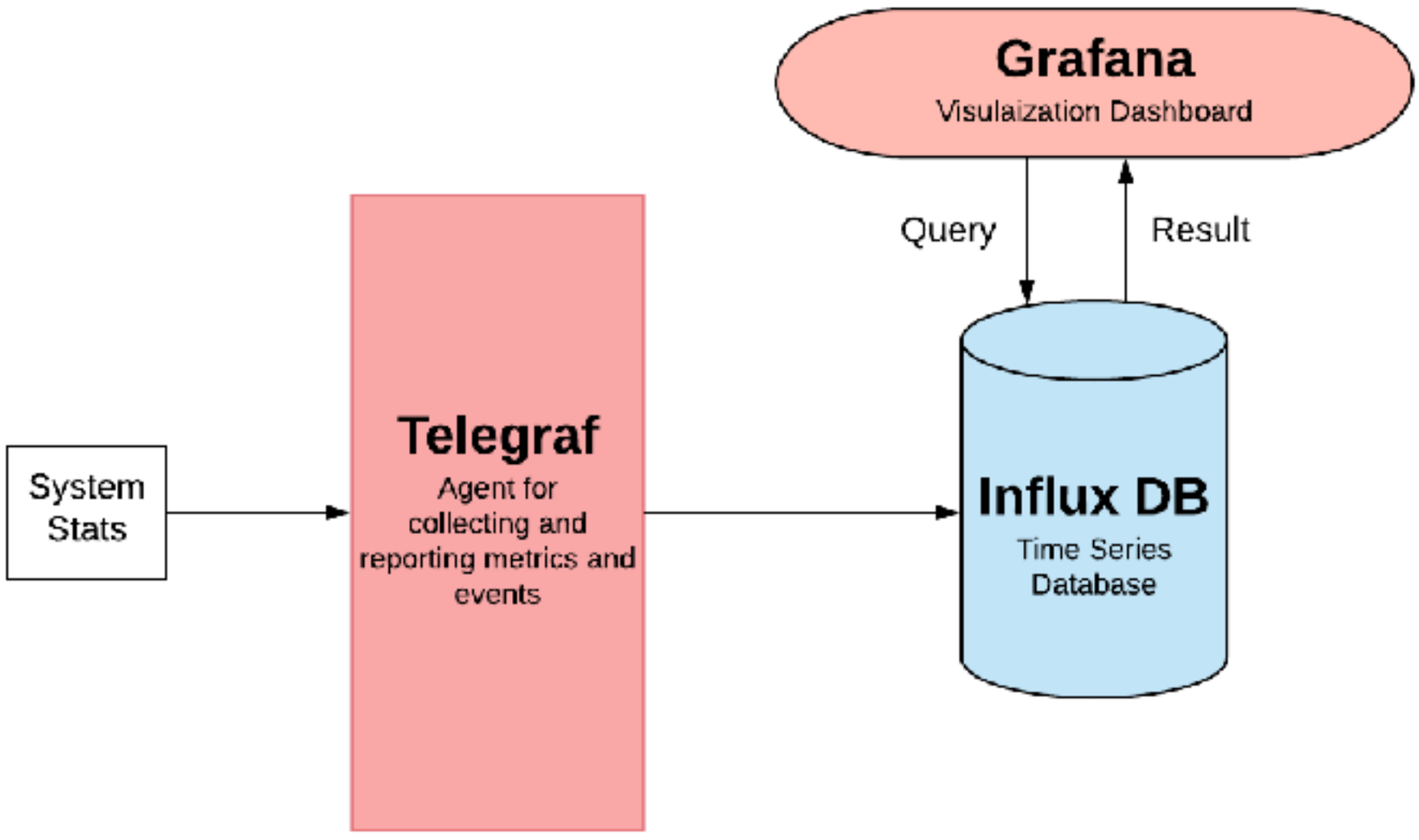
GO JEK

Writing

Would you like to know if there is a

Memory leak

Metric Tracking Architecture



Initialise Statsd

```
InitiateStatsDMetrics(config StatsDConfig)  {
  address := fmt.Sprintf("%s:%s", host, port)
  statsD, err := statsdv2.New(statsdv2.Address(address),
statsdv2.Prefix(appname))
}
```

StatsD Middleware

```
func StatsDMiddlewareLogger() negroni.HandlerFunc {
    return negroni.HandlerFunc(func(rw http.ResponseWriter, r *http.Request, next
http.HandlerFunc) {
        vars := mux.Vars(r)
        path := r.URL.Path
        key := GetKeyStructure(r.URL.Path)
        t := TimingInStatsD()
        next(rw, r)
        noOfGoRoutine := runtime.NumGoroutine()
        SendInStatsD(key+".time", t)
        IncrementInStatsD(key + ".calls")
        GaugeKeyInStatsD(key+".goroutines", noOfGoRoutine)
    })
}
```

StatsD Middleware

```
func StatsDMiddlewareLogger() negroni.HandlerFunc {
    return negroni.HandlerFunc(func(rw http.ResponseWriter, r *http.Request, next
http.HandlerFunc) {
        vars := mux.Vars(r)
        path := r.URL.Path
        key := GetKeyStructure(r.URL.Path)
        t := TimingInStatsD()
        next(rw, r)
        noOfGoRoutine := runtime.NumGoroutine()
        SendInStatsD(key+".time", t)
        IncrementInStatsD(key + ".calls")
        GaugeKeyInStatsD(key+".goroutines", noOfGoRoutine)
    })
}
```

StatsD Middleware

```
func StatsDMiddlewareLogger() negroni.HandlerFunc {
    return negroni.HandlerFunc(func(rw http.ResponseWriter, r *http.Request, next
http.HandlerFunc) {
        vars := mux.Vars(r)
        path := r.URL.Path
        key := GetKeyStructure(r.URL.Path)
        t := TimingInStatsD()
        next(rw, r)
        noOfGoRoutine := runtime.NumGoroutine()
        SendInStatsD(key+".time", t)
        IncrementInStatsD(key + ".calls")
        GaugeKeyInStatsD(key+".goroutines", noOfGoRoutine)
    })
}
```

StatsD Middleware

```
func StatsDMiddlewareLogger() negroni.HandlerFunc {
    return negroni.HandlerFunc(func(rw http.ResponseWriter, r *http.Request, next
http.HandlerFunc) {
        vars := mux.Vars(r)
        path := r.URL.Path
        key := GetKeyStructure(r.URL.Path)
        t := TimingInStatsD()
        next(rw, r)
        noOfGoRoutine := runtime.NumGoroutine()
        SendInStatsD(key+".time", t)
        IncrementInStatsD(key + ".calls")
        GaugeKeyInStatsD(key+".goroutines", noOfGoRoutine)
    })
}
```

StatsD Middleware

```
func StatsDMiddlewareLogger() negroni.HandlerFunc {
    return negroni.HandlerFunc(func(rw http.ResponseWriter, r *http.Request, next
http.HandlerFunc) {
        vars := mux.Vars(r)
        path := r.URL.Path
        key := GetKeyStructure(r.URL.Path)
        t := TimingInStatsD()
        next(rw, r)
        noOfGoRoutine := runtime.NumGoroutine()
        SendInStatsD(key+".time", t)
        IncrementInStatsD(key + ".calls")
        GaugeKeyInStatsD(key+".goroutines", noOfGoRoutine)
    })
}
```

StatsD Middleware

```
func StatsDMiddlewareLogger() negroni.HandlerFunc {
    return negroni.HandlerFunc(func(rw http.ResponseWriter, r *http.Request, next
http.HandlerFunc) {
        vars := mux.Vars(r)
        path := r.URL.Path
        key := GetKeyStructure(r.URL.Path)
        t := TimingInStatsD()
        next(rw, r)
        noOfGoRoutine := runtime.NumGoroutine()
        SendInStatsD(key+".time", t)
        IncrementInStatsD(key + ".calls")
        GaugeKeyInStatsD(key+".goroutines", noOfGoRoutine)
    })
}
```

StatsD Middleware

```
func StatsDMiddlewareLogger() negroni.HandlerFunc {
    return negroni.HandlerFunc(func(rw http.ResponseWriter, r *http.Request, next
http.HandlerFunc) {
        vars := mux.Vars(r)
        path := r.URL.Path
        key := GetKeyStructure(r.URL.Path)
        t := TimingInStatsD()
        next(rw, r)
        noOfGoRoutine := runtime.NumGoroutine()
        SendInStatsD(key+".time", t)
        IncrementInStatsD(key + ".calls")
        GaugeKeyInStatsD(key+".goroutines", noOfGoRoutine)
    })
}
```

StatsD Middleware

```
func StatsDMiddlewareLogger() negroni.HandlerFunc {
    return negroni.HandlerFunc(func(rw http.ResponseWriter, r *http.Request, next
http.HandlerFunc) {
        vars := mux.Vars(r)
        path := r.URL.Path
        key := GetKeyStructure(r.URL.Path)
        t := TimingInStatsD()
        next(rw, r)
        noOfGoRoutine := runtime.NumGoroutine()
        SendInStatsD(key+".time", t)
        IncrementInStatsD(key + ".calls")
        GaugeKeyInStatsD(key+".goroutines", noOfGoRoutine)
    })
}
```

StatsD Middleware

```
func TimingInStatsD()
*statsdv2.Timing {
if statsD == nil {
return nil
}
t := statsD.NewTiming()
return &t
}
```

```
func IncrementInStatsD(key string) bool {
if statsD == nil {
return false
}
statsD.Increment(key)
return true
}
```

```
func SendInStatsD(key string, t *statsdv2.Timing) bool {
if statsD == nil {
return false
}
t.Send(key)
return true
}
```

```
func GetKeyStructure(url string) string {
baseKey := "go.response"
basePath := strings.Split(url, "/GF")[0]
keyBasePath := strings.Replace(basePath, "/",
".", len(basePath))
key := baseKey + keyBasePath
return key
}
```

StatsD Middleware

```
func TimingInStatsD()
*statsdv2.Timing {
if statsD == nil {
return nil
}
t := statsD.NewTiming()
return &t
}
```

```
func IncrementInStatsD(key string) bool {
if statsD == nil {
return false
}
statsD.Increment(key)
return true
}
```

```
func SendInStatsD(key string, t *statsdv2.Timing) bool {
if statsD == nil {
return false
}
t.Send(key)
return true
}
```

```
func GetKeyStructure(url string) string {
baseKey := "go.response"
basePath := strings.Split(url, "/GF")[0]
keyBasePath := strings.Replace(basePath, "/",
".", len(basePath))
key := baseKey + keyBasePath
return key
}
```

StatsD Middleware

```
func TimingInStatsD()
*statsdv2.Timing {
if statsD == nil {
return nil
}
t := statsD.NewTiming()
return &t
}
```

```
func SendInStatsD(key string, t *statsdv2.Timing) bool {
if statsD == nil {
return false
}
t.Send(key)
return true
}
```

```
func IncrementInStatsD(key string) bool {
if statsD == nil {
return false
}
statsD.Increment(key)
return true
}
```

```
func GetKeyStructure(url string) string {
baseKey := "go.response"
basePath := strings.Split(url, "/GF")[0]
keyBasePath := strings.Replace(basePath, "/",
".", len(basePath))
key := baseKey + keyBasePath
return key
}
```

StatsD Middleware

```
func TimingInStatsD()
*statsdv2.Timing {
if statsD == nil {
return nil
}
t := statsD.NewTiming()
return &t
}
```

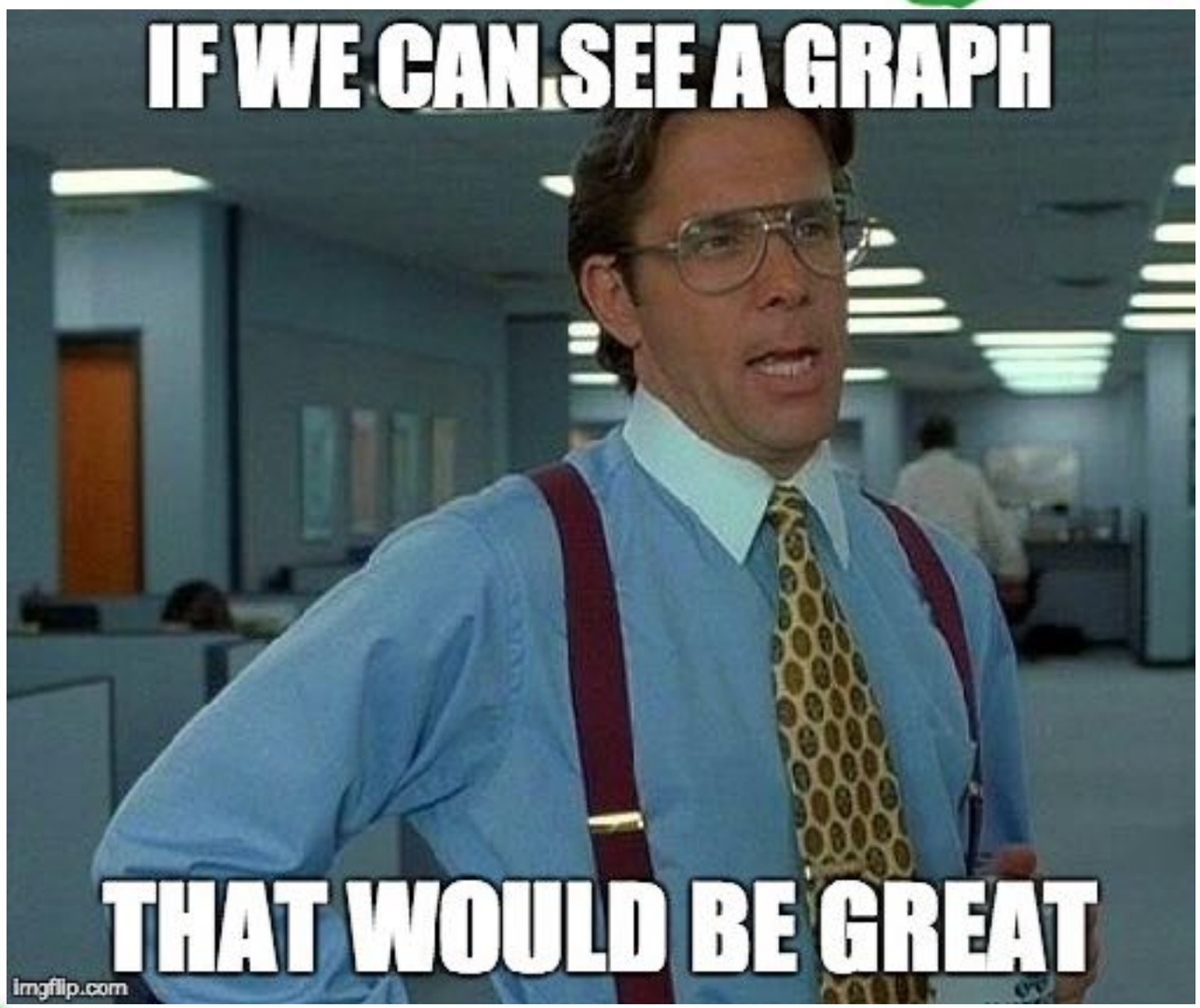
```
func SendInStatsD(key string, t *statsdv2.Timing) bool {
if statsD == nil {
return false
}
t.Send(key)
return true
}
```

```
func IncrementInStatsD(key string) bool {
if statsD == nil {
return false
}
statsD.Increment(key)
return true
}
```

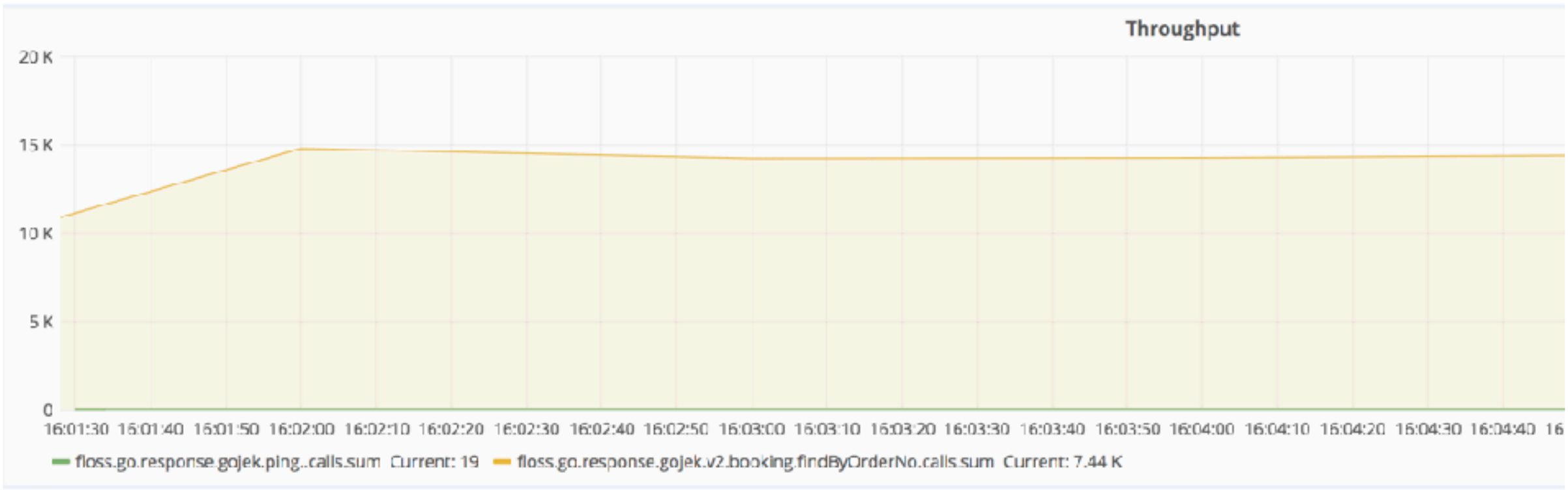
```
func GetKeyStructure(url string) string {
baseKey := "go.response"
basePath := strings.Split(url, "/GF")[0]
keyBasePath := strings.Replace(basePath, "/",
".", len(basePath))
key := baseKey + keyBasePath
return key
}
```

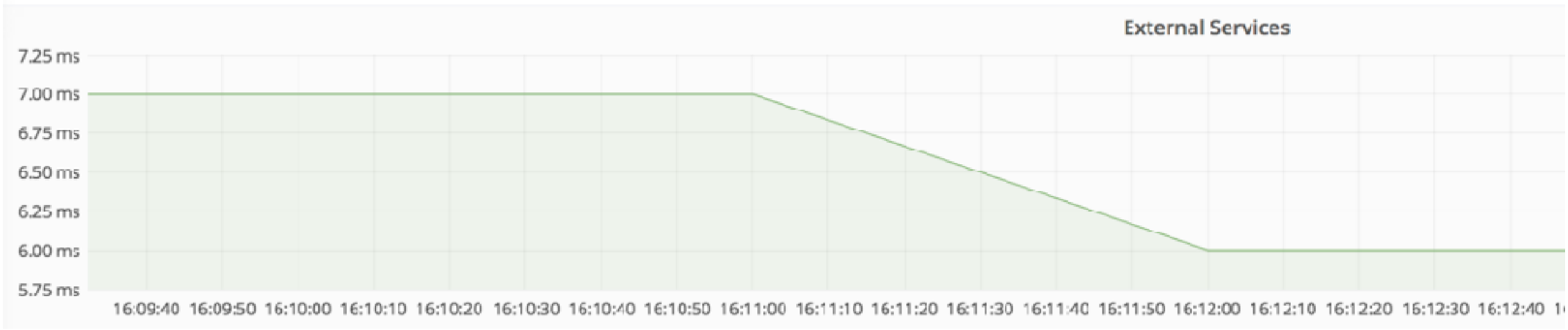
StatsD Middleware

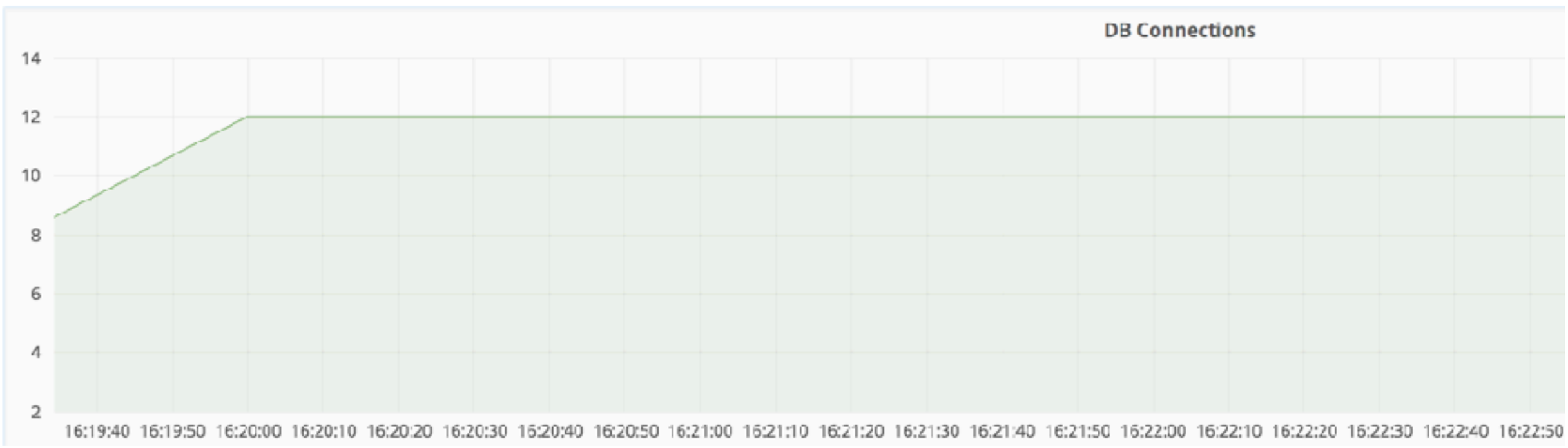
```
n := negroni.New(negroni.NewRecovery())
n.Use(instrumentation.StatsDMiddlewareLogger())
```


IF WE CAN SEE A GRAPH
THAT WOULD BE GREAT
imgflip.com

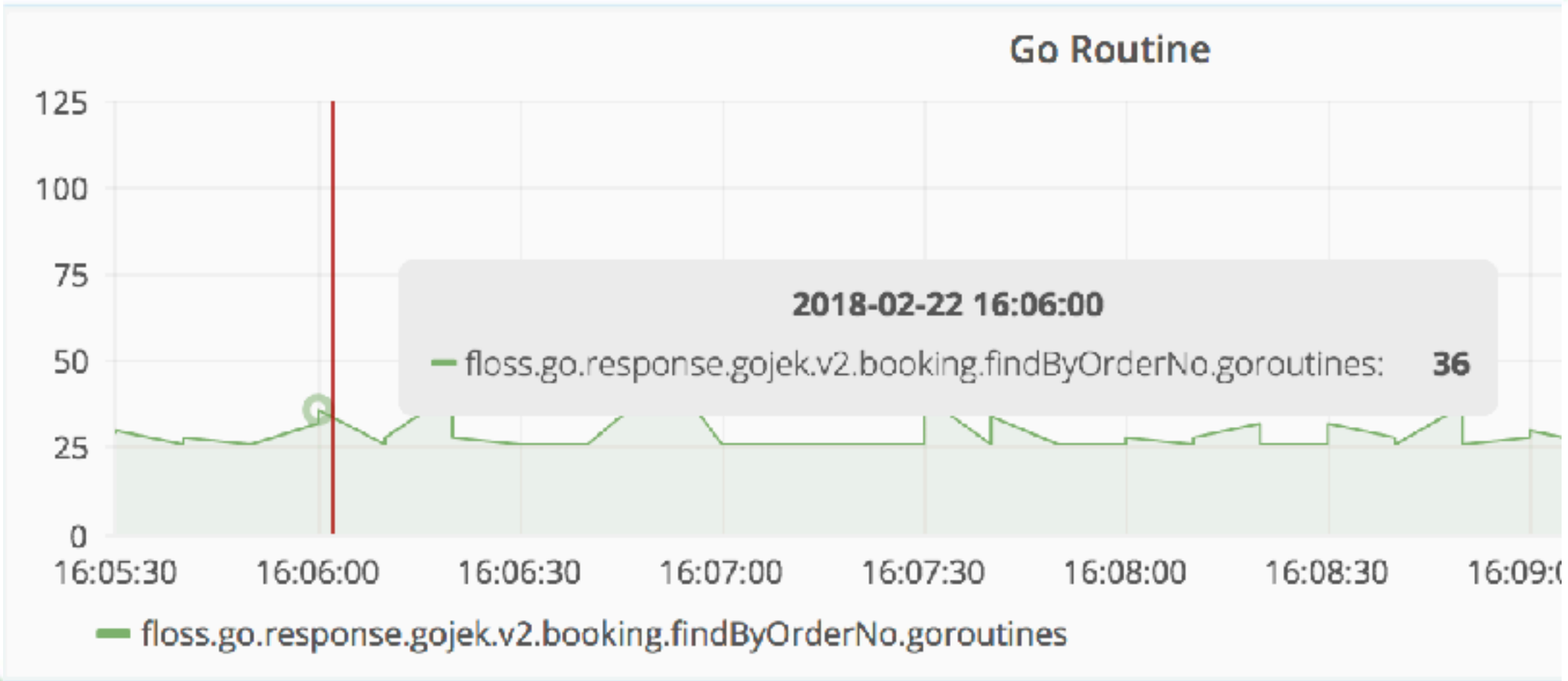
GO-JEK


Throughput
20 K
15 K
10 K
5 K
0
16:01:30 16:01:40 16:01:50 16:02:00 16:02:10 16:02:20 16:02:30 16:02:40 16:02:50 16:03:00 16:03:10 16:03:20 16:03:30 16:03:40 16:03:50 16:04:00 16:04:10 16:04:20 16:04:30 16:04:40 16
floss.go.response.gojek.ping..calls.sum Current: 19
floss.go.response.gojek.v2.booking.findByOrderNo.calls.sum Current: 7.44 K


External Services
7.25 ms
7.00 ms
6.75 ms
6.50 ms
6.25 ms
6.00 ms
5.75 ms
16:09:40
16:09:50
16:10:00
16:10:10
16:10:20
16:10:30
16:10:40
16:10:50
16:11:00
16:11:10
16:11:20
16:11:30
16:11:40
16:11:50
16:12:00
16:12:10
16:12:20
16:12:30
16:12:40


DB Connections
14
12
10
8
6
4
2
16:19:40
16:19:50
16:20:00
16:20:10
16:20:20
16:20:30
16:20:40
16:20:50
16:21:00
16:21:10
16:21:20
16:21:30
16:21:40
16:21:50
16:22:00
16:22:10
16:22:20
16:22:30
16:22:40
16:22:50

GO JEK


Go Routine
125
100
75
50
25
0
16:05:30
16:06:00
16:06:30
16:07:00
16:07:30
16:08:00
16:08:30
2018-02-22 16:06:00
floss.go.response.gojek.v2.booking.findByOrderNo.goroutines: 36
floss.go.response.gojek.v2.booking.findByOrderNo.goroutines

New Relic Instrumentations

```
func StartDataSegmentNowForDataStore(op string, tableName string,
txn newrelic.Transaction,datastore newrelic.DatastoreProduct)
newrelic.DatastoreSegment {
        s := newrelic.DatastoreSegment{
            Product:    datastore,
            Collection: tableName,
            Operation:  op,
        }

        s.StartTime = newrelic.StartSegmentNow(txn)
        return s
}
```

New Relic Instrumentations

```
func StartDataSegmentNowForDataStore(op string, tableName string,
txn newrelic.Transaction,datastore newrelic.DatastoreProduct)
newrelic.DatastoreSegment {
        s := newrelic.DatastoreSegment{
            Product:    datastore,
            Collection: tableName,
            Operation:  op,
        }

        s.StartTime = newrelic.StartSegmentNow(txn)
        return s
}
```

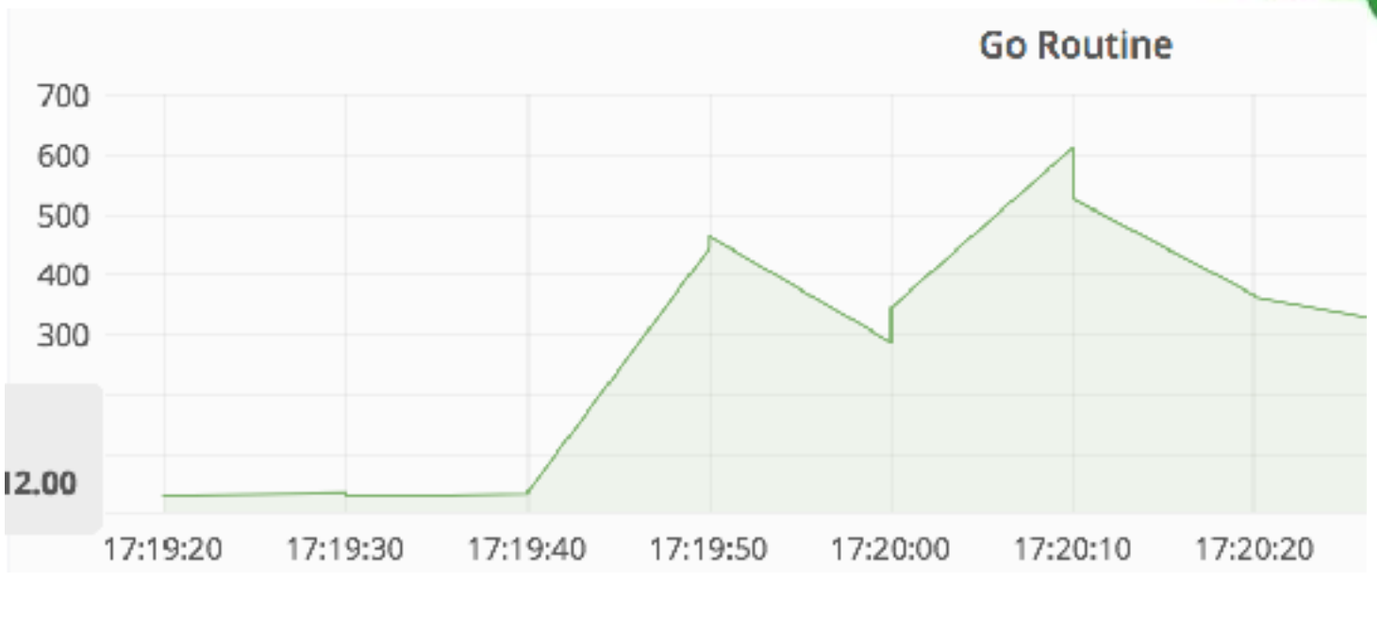
New Relic Instrumentations

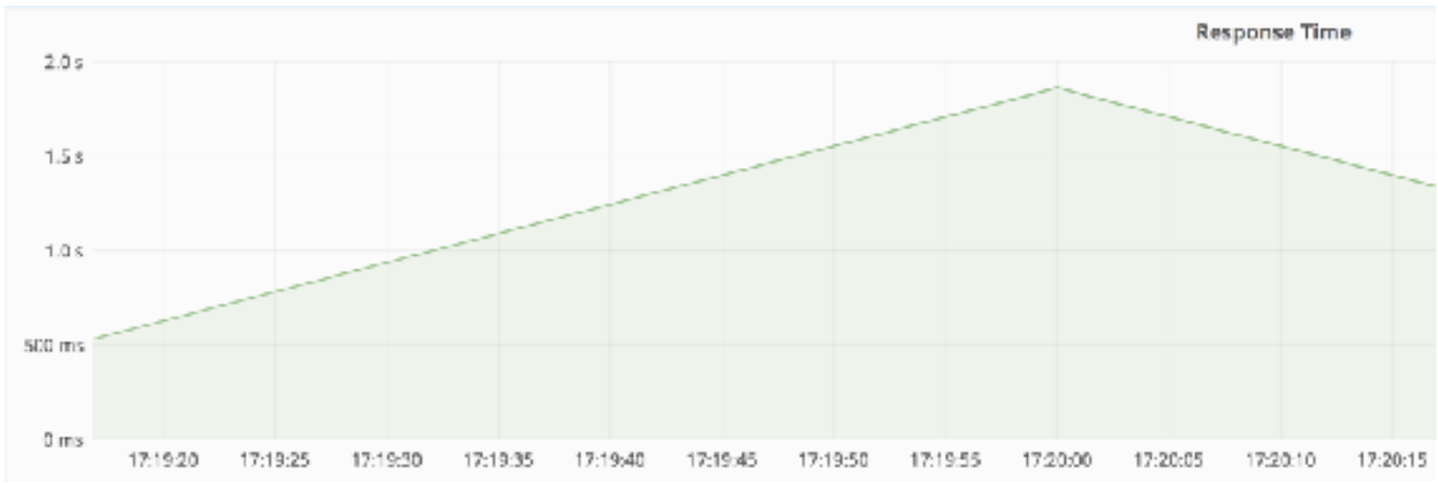
```
func StartDataSegmentNowForDataStore(op string, tableName string,
txn newrelic.Transaction,datastore newrelic.DatastoreProduct)
newrelic.DatastoreSegment {
        s := newrelic.DatastoreSegment{
            Product:    datastore,
            Collection: tableName,
            Operation:  op,
        }

        s.StartTime = newrelic.StartSegmentNow(txn)
        return s
}
```

Name	Vendor	Operation	Collection	Call count	Throughput (cpm)	Avg (ms)	Min (ms)	Max (ms)
MySQL findBookingQuery	MySQL	findBookingQuery	(all)	125,000	4,150	0.025	0.001	1.89
MySQL Booking findBookingQuery	MySQL	findBookingQuery	(all)	125,000	4,150	0.025	0.001	1.89
MySQL findRouteItemRepository	MySQL	findRouteItemRepository	(all)	114,000	3,800	0.008	0.001	1.73
MySQL route_item findRouteItemRepository	MySQL	findRouteItemRepository	(all)	114,000	3,800	0.008	0.001	1.73
MySQL findAddressQuery	MySQL	findAddressQuery	(all)	113,000	3,760	0.006	0.001	1.69
MySQL booking_route findAddressQuery	MySQL	findAddressQuery	(all)	113,000	3,760	0.006	0.001	1.69
MySQL booking_driver_cancellation_log findBookingDriverCancellationQuery	MySQL	findBookingDriverCancellationQuery	(all)	114,000	3,790	0.006	0.001	1.97
MySQL findBookingDriverCancellationQuery	MySQL	findBookingDriverCancellationQuery	(all)	114,000	3,790	0.006	0.001	1.97
MySQL pricing_info findPriceInfoQuery	MySQL	findPriceInfoQuery	(all)	20,600	686	0.003	0.001	0.903
MySQL findPriceInfoQuery	MySQL	findPriceInfoQuery	(all)	20,600	686	0.003	0.001	0.903

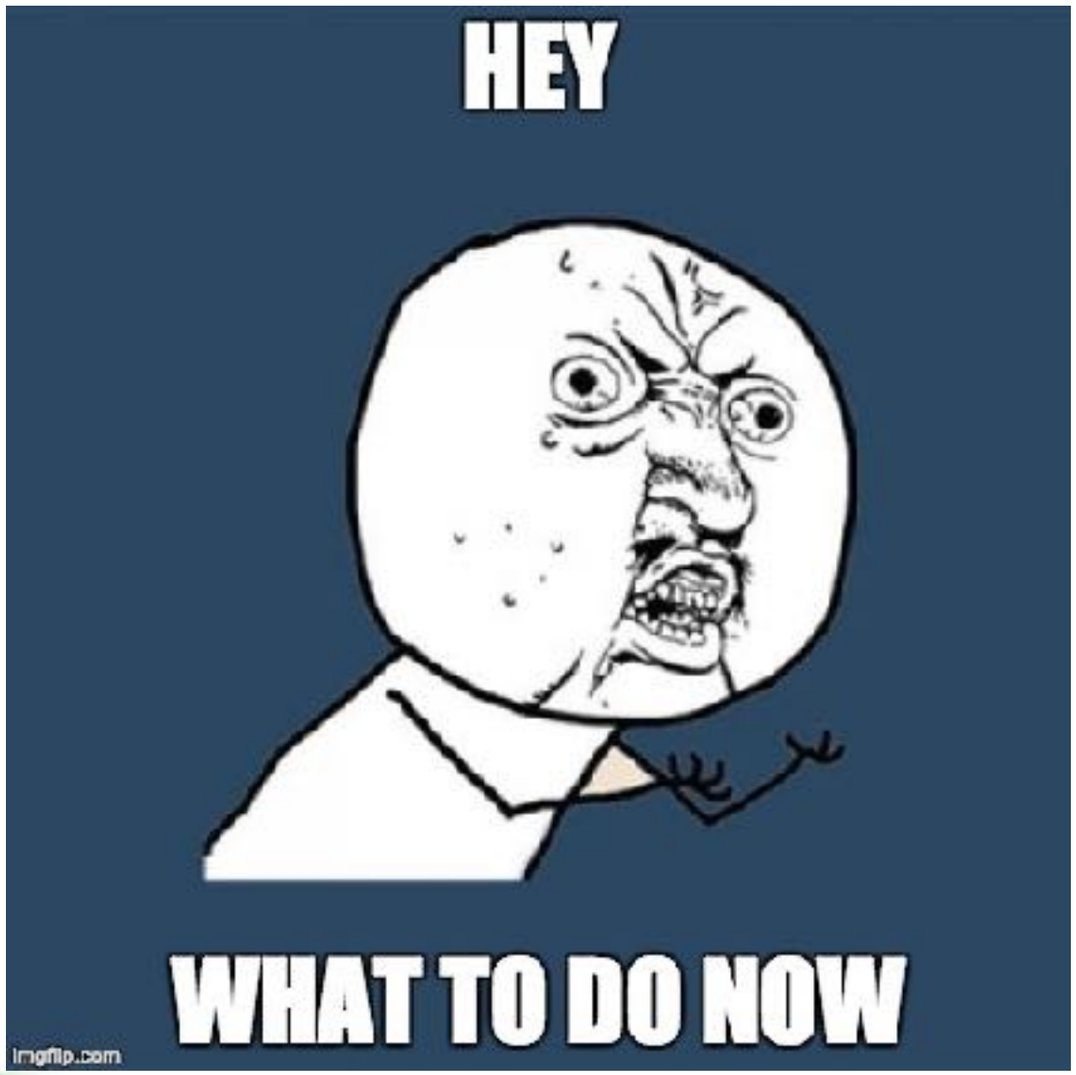
GO JEK

GO JEK


Go Routine
700
600
500
400
300
12.00
17:19:20
17:19:30
17:19:40
17:19:50
17:20:00
17:20:10
17:20:20


Response Time
2.0 s
1.5 s
1.0 s
500 ms
0 ms
17:19:20
17:19:25
17:19:30
17:19:35
17:19:40
17:19:45
17:19:50
17:19:55
17:20:00
17:20:05
17:20:10
17:20:15

GO JEK



```go
import "net/http/pprof"

func Router(dependencies *service.Instances) *mux.Router {
    router := mux.NewRouter()
    router.HandleFunc("/debug/pprof/", pprof.Index)
    router.HandleFunc("/debug/pprof/cmdline", pprof.Cmdline)
    router.HandleFunc("/debug/pprof/profile", pprof.Profile)
    router.HandleFunc("/debug/pprof/symbol", pprof.Symbol)
    router.HandleFunc("/debug/pprof/trace", pprof.Trace)
    router.HandleFunc("/debug/pprof/goroutine", pprof.Index)
    return router
}
```

```
import "net/http/pprof"

func Router(dependencies *service.Instances) *mux.Router {
    router := mux.NewRouter()
    router.HandleFunc("/debug/pprof/", pprof.Index)
    router.HandleFunc("/debug/pprof/cmdline", pprof.Cmdline)
    router.HandleFunc("/debug/pprof/profile", pprof.Profile)
    router.HandleFunc("/debug/pprof/symbol", pprof.Symbol)
    router.HandleFunc("/debug/pprof/trace", pprof.Trace)
    router.HandleFunc("/debug/pprof/goroutine", pprof.Index)
    return router
}
```

/debug/pprof/

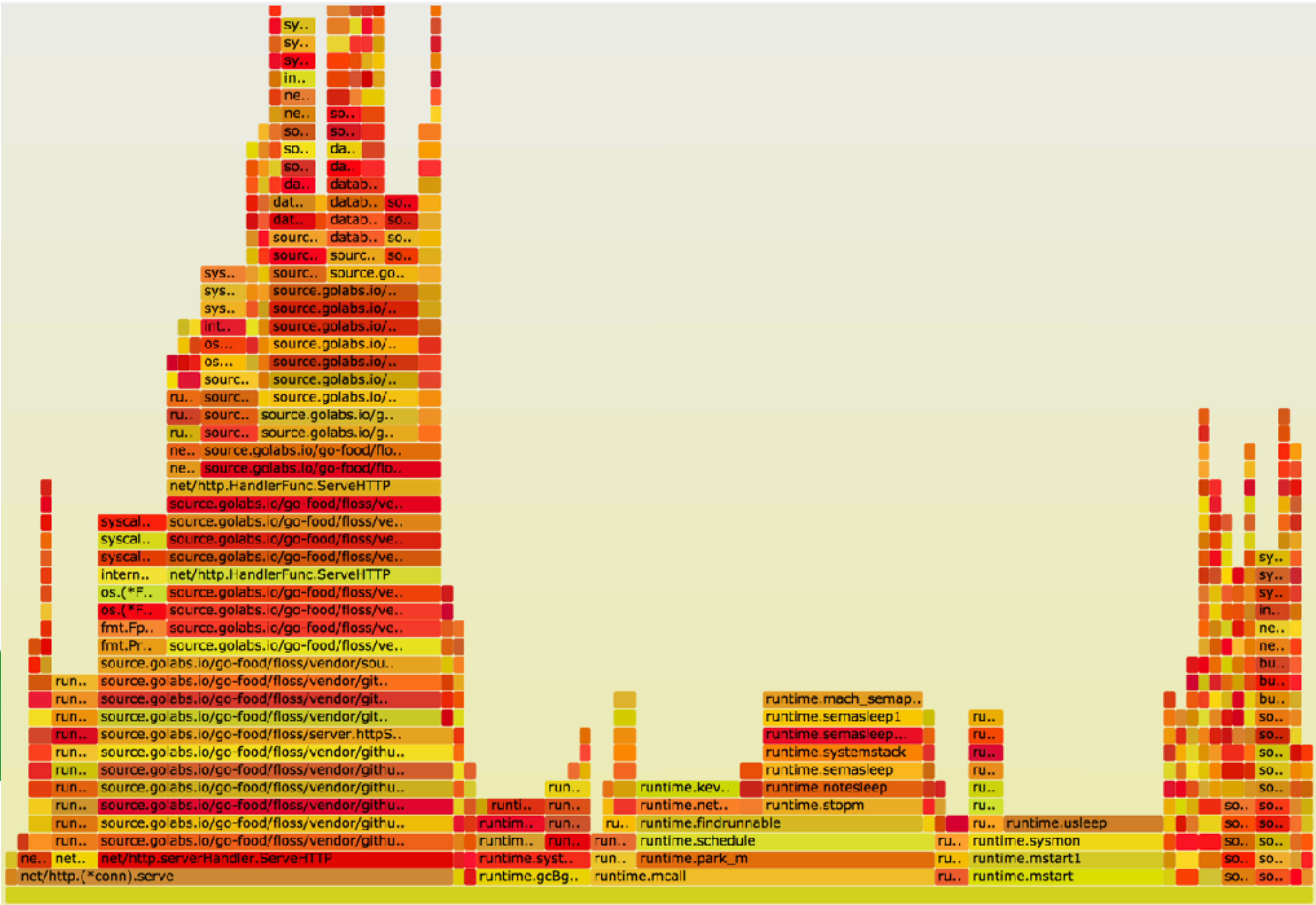
profiles:

0	block
799	goroutine
72	heap
0	mutex
13	threadcreate

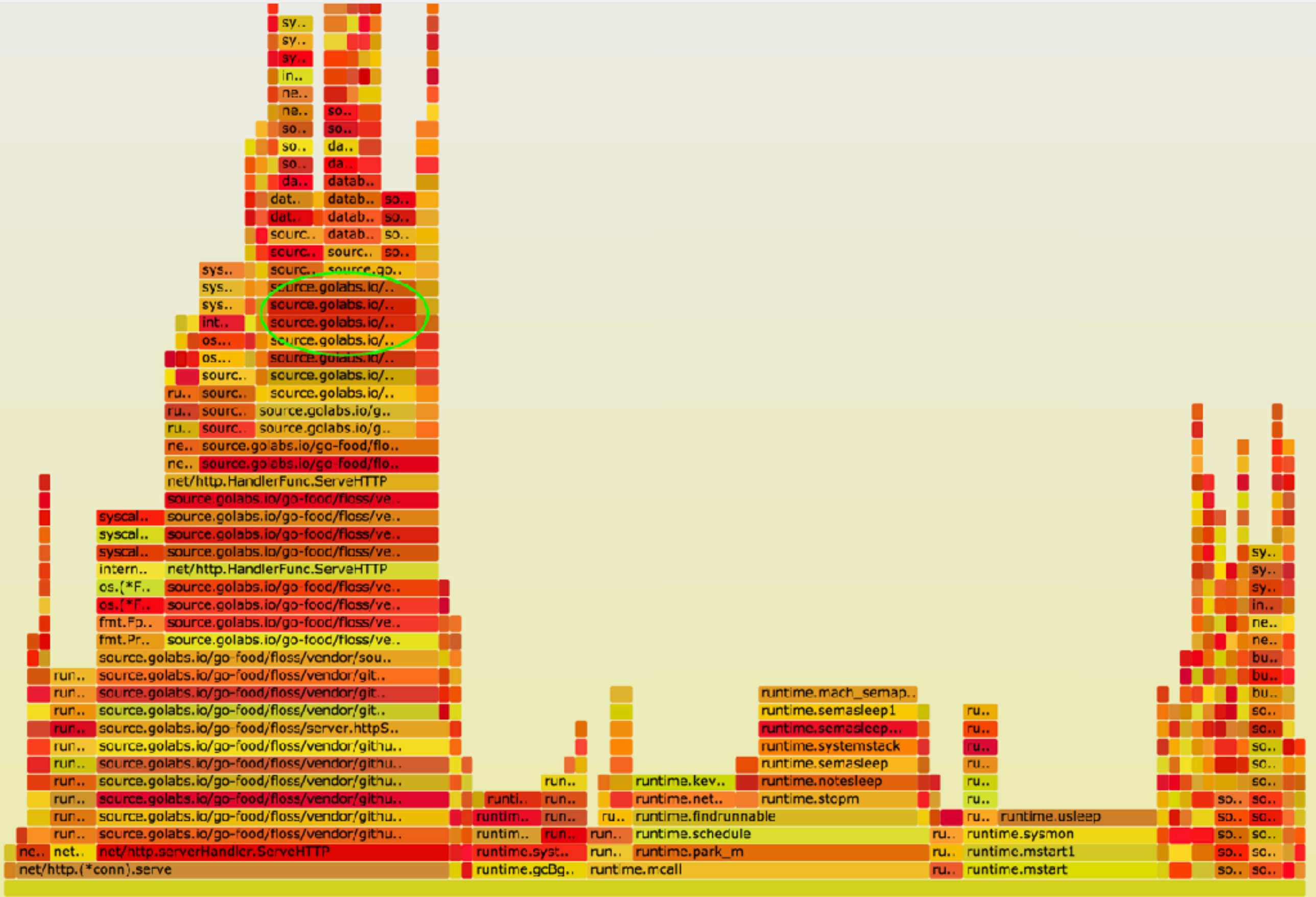
full goroutine stack dump

```
goroutine profile: total 7945
3959 @ 0x102f4ec 0x1035e9 0x13aa521 0x13ac8c6 0x13ac602 0x13ac822 0x13dfc33 0x1548b0f 0x1564b67 0x15664cd 0x1569039 0x156bea3 0x1569533 0x1569190 0x156c26e 0x156e3a5 0x156510e 0x156eb19 0x133a7fb 0x128ac14
0x155acf1 0x155b419 0x1538cbd 0x1573168 0x128ac14 0x1536ddd 0x15353be 0x1535475 0x1537c20 0x1549fd7 0x15353be 0x1535475
#       0x13aa520       database/sql.(*DB).conn+0x6f0                                                   /usr/local/opt/go/libexec/src/database/sql/sql.go:970
#       0x13ac8c5       database/sql.(*DB).query+0x65
/usr/local/opt/go/libexec/src/database/sql/sql.go:1299
#       0x13ac601       database/sql.(*DB).QueryContext+0xd1
/usr/local/opt/go/libexec/src/database/sql/sql.go:1281
#       0x13ac821       database/sql.(*DB).Query+0x81
/usr/local/opt/go/libexec/src/database/sql/sql.go:1295
#       0x13dfc32       source.golabs.io/go-food/floss/vendor/github.com/jmoiron/sqlx.(*DB).QueryRowx+0x72       /Users/blackvirus/Go/src/source.golabs.io/go-
food/floss/vendor/github.com/jmoiron/sqlx/sqlx.go:355
#       0x1548b0e       source.golabs.io/go-food/floss/vendor/source.golabs.io/go-food/food-utils-library-go.(*BaseRepository).MakeRowDbCall+0x6e       /Users/blackvirus/Go/src/source.golabs.io/go-
food/floss/vendor/source.golabs.io/go-food/food-utils-library-go/base_repository.go:75
#       0x1564b66       source.golabs.io/go-food/floss/repository.bookingRepository.Get+0xd6       /Users/blackvirus/Go/src/source.golabs.io/go-
food/floss/repository/booking.go:27
#       0x15664cc       source.golabs.io/go-food/floss/repository.(*bookingRepository).Get+0x7c       <autogenerated>:1
#       0x1569038       source.golabs.io/go-food/floss/service.bookingService.Fetch+0x68       /Users/blackvirus/Go/src/source.golabs.io/go-
food/floss/service/booking_service.go:19
#       0x156bea2       source.golabs.io/go-food/floss/service.(*bookingService).Fetch+0x72       <autogenerated>:1
#       0x1569532       source.golabs.io/go-food/floss/service.findByOrderNumberService.fetchBookingFromBookingService+0x62       /Users/blackvirus/Go/src/source.golabs.io/go-
food/floss/service/find_by_order_number_service.go:94
#       0x156918f       source.golabs.io/go-food/floss/service.findByOrderNumberService.Get+0x7f       /Users/blackvirus/Go/src/source.golabs.io/go-
food/floss/service/find_by_order_number_service.go:76
#       0x156c26d       source.golabs.io/go-food/floss/service.(*findByOrderNumberService).Get+0x7d       <autogenerated>:1
#       0x156e3a4       source.golabs.io/go-food/floss/handler.GetOrderFromDB.func1+0x44       /Users/blackvirus/Go/src/source.golabs.io/go-
food/floss/handler/find_by_order_number.go:33
#       0x156510d       source.golabs.io/go-food/floss/repository.(*cache).FetchWithCache+0x2bd       /Users/blackvirus/Go/src/source.golabs.io/go-
food/floss/repository/cache.go:46
#       0x156eb18       source.golabs.io/go-food/floss/handler.FindOrderByOrderNumber.func1+0x478       /Users/blackvirus/Go/src/source.golabs.io/go-
food/floss/handler/find_by_order_number.go:66
#       0x133a7fa       source.golabs.io/go-food/floss/vendor/github.com/getsentry/raven-go.RecoveryHandler.func1+0x8a       /Users/blackvirus/Go/src/source.golabs.io/go-
food/floss/vendor/github.com/getsentry/raven-go/http.go:82
#       0x128ac13       net/http.HandlerFunc.ServeHTTP+0x43       /usr/local/opt/go/libexec/src/net/http/server.go:1918
#       0x155acf0       source.golabs.io/go-food/floss/vendor/github.com/newrelic/go-agent/_integrations/nrgorilla/v1.instrumentedHandler.ServeHTTP+0x120       /Users/blackvirus/Go/src/source.golabs.io/go-
food/floss/vendor/github.com/newrelic/go-agent/_integrations/nrgorilla/v1/nrgorilla.go:25
#       0x155b418       source.golabs.io/go-food/floss/vendor/github.com/newrelic/go-agent/_integrations/nrgorilla/v1.(*instrumentedHandler).ServeHTTP+0x68       <autogenerated>:1
#       0x1538cbc       source.golabs.io/go-food/floss/vendor/github.com/gorilla/mux.(*Router).ServeHTTP+0xec       /Users/blackvirus/Go/src/source.golabs.io/go-
food/floss/vendor/github.com/gorilla/mux/mux.go:159
#       0x1573167       source.golabs.io/go-food/floss/vendor/github.com/gorilla/mux.(*Router).ServeHTTP-fm+0x47       /Users/blackvirus/Go/src/source.golabs.io/go-
food/floss/server/api.go:54
#       0x128ac13       net/http.HandlerFunc.ServeHTTP+0x43       /usr/local/opt/go/libexec/src/net/http/server.go:1918
#       0x1536ddc       source.golabs.io/go-food/floss/vendor/github.com/codegangsta/negroni.Wrap.func1+0x1c       /Users/blackvirus/Go/src/source.golabs.io/go-
food/floss/vendor/github.com/codegangsta/negroni/negroni.go:46
#       0x15353bd       source.golabs.io/go-food/floss/vendor/github.com/codegangsta/negroni.HandlerFunc.ServeHTTP+0x4d       /Users/blackvirus/Go/src/source.golabs.io/go-
food/floss/vendor/github.com/codegangsta/negroni/negroni.go:29
#       0x1535474       source.golabs.io/go-food/floss/vendor/github.com/codegangsta/negroni.middleware.ServeHTTP+0xa4       /Users/blackvirus/Go/src/source.golabs.io/go-
food/floss/vendor/github.com/codegangsta/negroni/negroni.go:38
#       0x1537c1f       source.golabs.io/go-food/floss/vendor/github.com/codegangsta/negroni.(middleware).ServeHTTP-fm+0x5f       /Users/blackvirus/Go/src/source.golabs.io/go-
food/floss/vendor/github.com/codegangsta/negroni/negroni.go:30
#       0x1549fd6       source.golabs.io/go-food/floss/vendor/source.golabs.io/go-food/food-utils-library-go.StatsDMiddlewareLogger.func1+0x396       /Users/blackvirus/Go/src/source.golabs.io/go-
food/floss/vendor/source.golabs.io/go-food/food-utils-library-go/statsd.go:135
#       0x15353bd       source.golabs.io/go-food/floss/vendor/github.com/codegangsta/negroni.HandlerFunc.ServeHTTP+0x4d       /Users/blackvirus/Go/src/source.golabs.io/go-
food/floss/vendor/github.com/codegangsta/negroni/negroni.go:29
#       0x1535474       source.golabs.io/go-food/floss/vendor/github.com/codegangsta/negroni.middleware.ServeHTTP+0xa4       /Users/blackvirus/Go/src/source.golabs.io/go-
food/floss/vendor/github.com/codegangsta/negroni/negroni.go:38
```

go tool pprof -raw -seconds 10 http://localhost:3000/debug/pprof/profile



go tool pprof -raw -seconds 10 http://localhost:3000/debug/pprof/profile



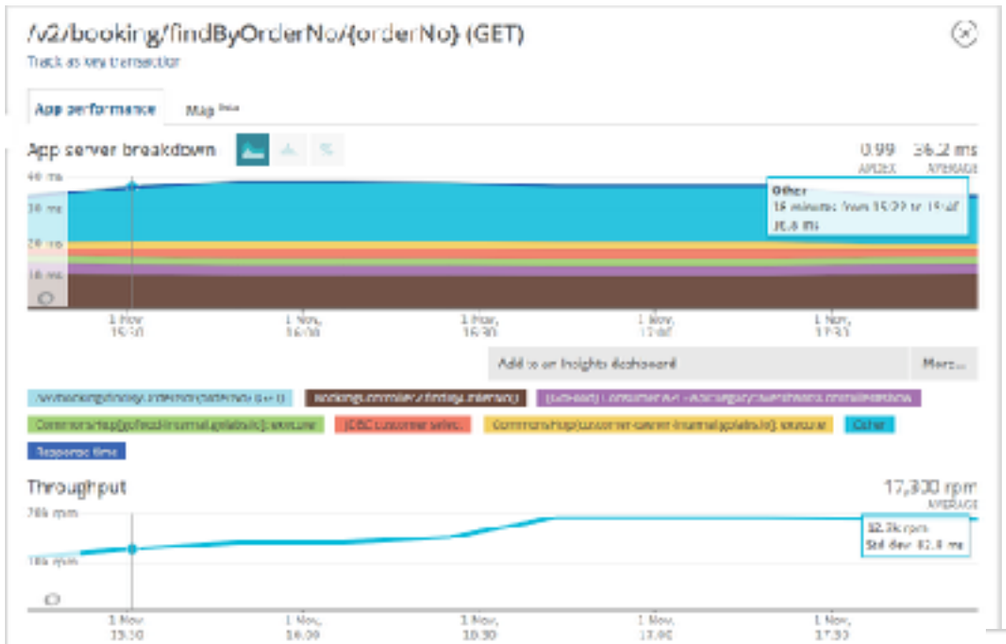
GO JEK

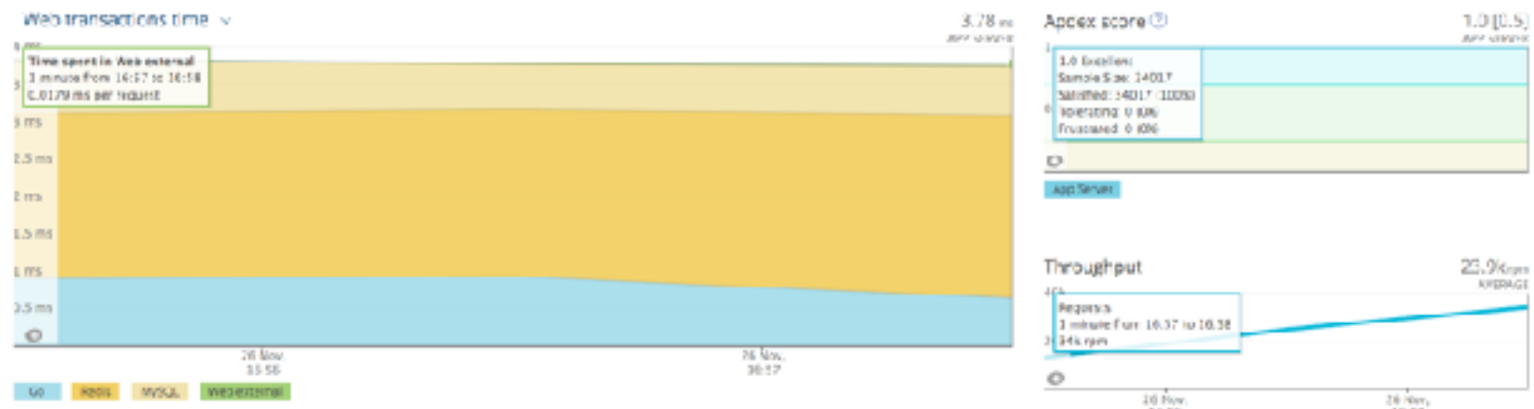
```
return rows.Next()
```

```
boolean := false
defer func() {
if rows != nil {
rows.Close()
}
}()
if rows.Next() {
boolean = true
}
return boolean, nil
```

GO JEK

Reduction in response times





Things to Remember

Metrics

Profiling

Monitoring

Things to Remember

Metrics

Profiling

Monitoring

Things to Remember

Metrics

Profiling

Monitoring

Thank You !

https://github.com/blackvirus18/gophercon2018

Linkedin: https://in.linkedin.com/in/deepeshnaini

Twitter: https://twitter.com/DeepeshNaini