Aurélien Rainone

Go Software Engineer

How to Instrument and Monitor a Go Application

The Student Hotel

11th December 2019

develer

- Is my program slow, can it go faster?
- How does my app perform in production?
- How does it perform under load?

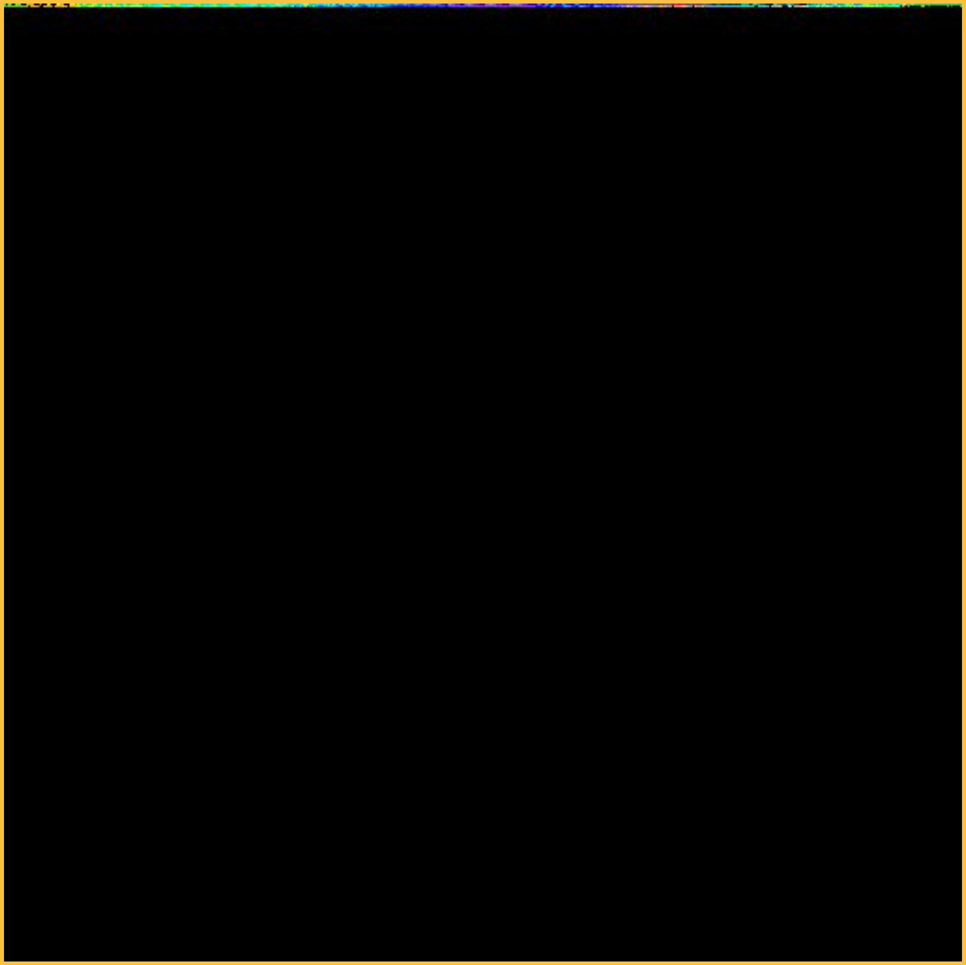
- Monitor a CLI application
    - Tracing
    - Profiling

- Monitor a HTTP server
    - Time Series Database
    - Metrics
    - Prometheus

Instrument a command line application

develer

Mandelbrot Set



```
$ git clone github.com/arl/monitoring
$ cd monitoring
$ cd fractal
```

```
for each frame:
    for each pixel:
        // c = x + yi
        c = complex(pixel.x, pixel.y)
        z = complex(0, 0)

        // does z ‘escapes’ to infinity
        for n=0 => max_iteration:
            z = z²+c
            if mod(z) > some_number:
                break

        // use n to color the pixel
        if n == max_iterations:
            pixel.color = black
        else:
            pixel.color = f(n)
```

https://en.wikipedia.org/wiki/Mandelbrot_set

develer

Execution Tracer

golang.org/pkg/runtime/trace/
go tool trace

- To spot concurrency problems
- See the interaction between our program and the Go runtime (scheduler, GC) and ultimately with the OS

Benchmarking

golang.org/pkg/testing/
go test -bench …
golang.org/x/perf/cmd/benchstat

- To measure performance at the function level
- Compare code performance

Profiling

golang.org/pkg/runtime/pprof/

go tool pprof -http=:

CPU, Memory, Contention, etc.

- Using the pprof package directly (Start/Stop)
- Adding a profiling HTTP handler to our program
- go test -cpuprofile or -memprofile (only CPU and Memory)

CPU Profiling

```
go test -bench . -run $^ -cpuprofile cpu.out
```

Stops your program every 10ms and records the stack trace of each goroutine.

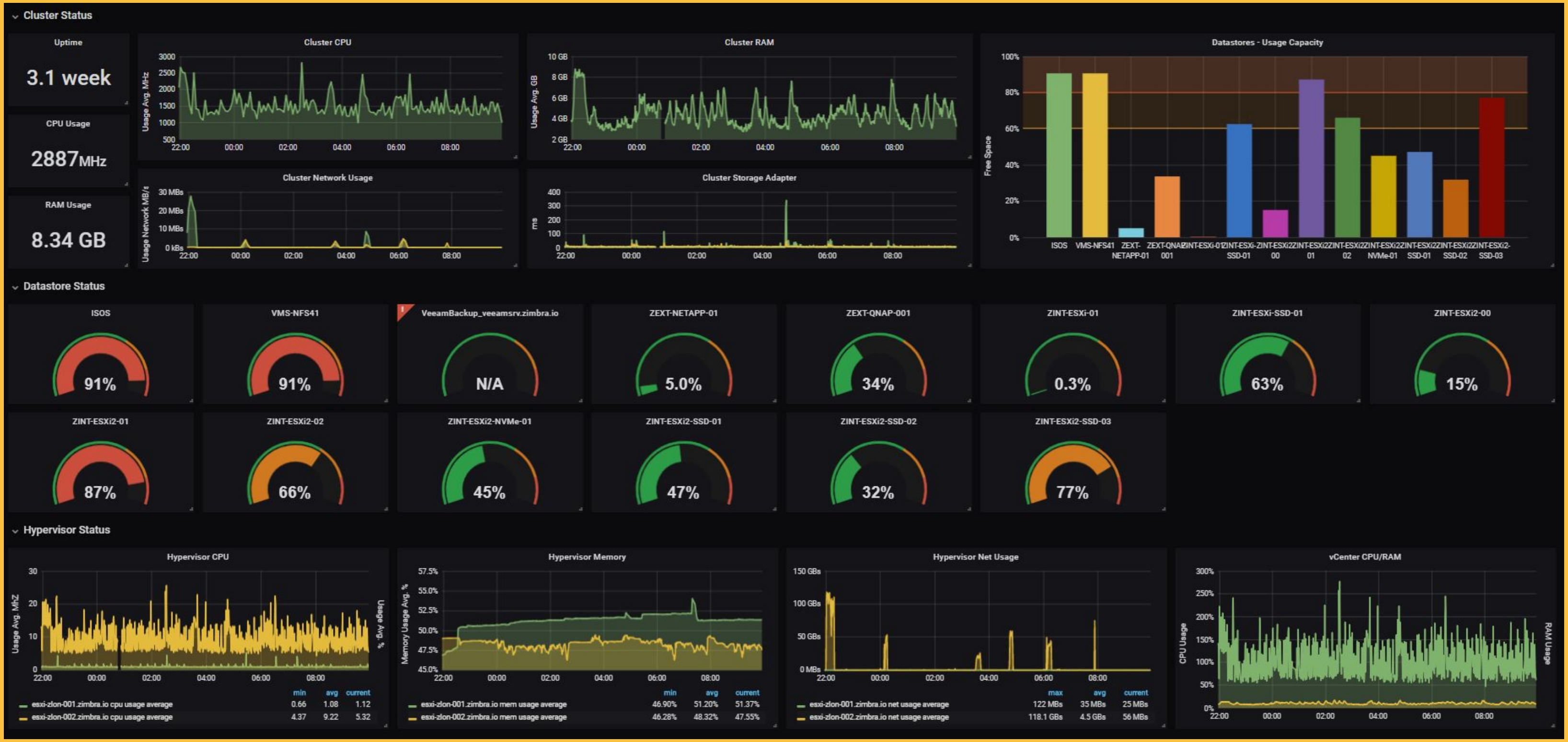
Memory Profiling

```
go test -bench . -run $^ -memprofile cpu.out
```

Records the stack trace at each heap allocation.

alloc_objects: objects allocated
alloc_space: bytes allocated

inuse_space: objects allocated and not GC’ed
inuse_objects: bytes allocated and not GC’ed



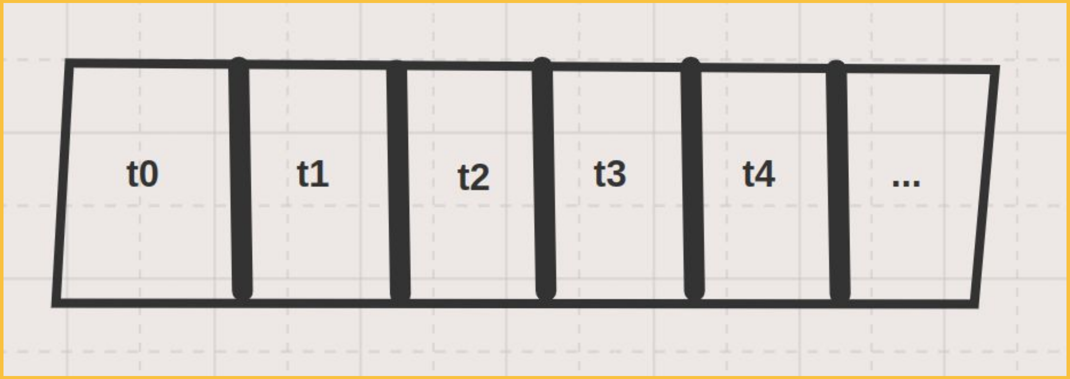
	min	avg	current
esxi-zlon-001.zimbra.io cpu usage average	0.66	1.08	1.12
esxi-zlon-002.zimbra.io cpu usage average	4.37	9.22	5.32

	min	avg	current
esxi-zlon-001.zimbra.io mem usage average	46.90%	51.20%	51.37%
esxi-zlon-002.zimbra.io mem usage average	46.28%	48.32%	47.55%

	max	avg	current
esxi-zlon-001.zimbra.io net usage average	122 MBs	35 MBs	25 MBs
esxi-zlon-002.zimbra.io net usage average	118.1 GBs	4.5 GBs	56 MBs

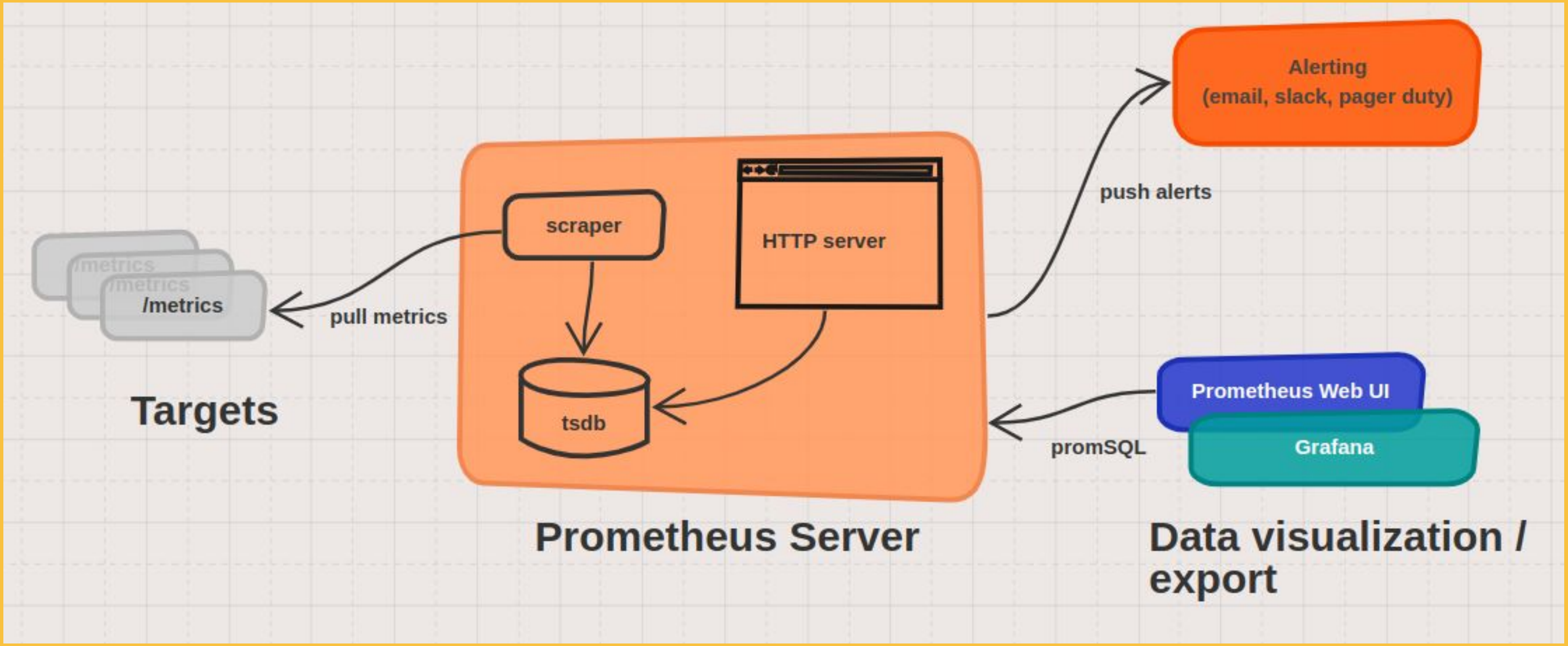
Service Level Metrics

Time Series Databases



Timestamp	Metric ID	Value
13:01:45	12	10327
13:01:47	12	10500
13:01:48	14	3.14159275

Prometheus


/metrics
/metrics
/metrics
pull metrics
Targets
scraper
tsdb
HTTP server
Prometheus Server
push alerts
Alerting
(email, slack, pager duty)
Prometheus Web UI
Grafana
promSQL
Data visualization / export

Metric types

Counter

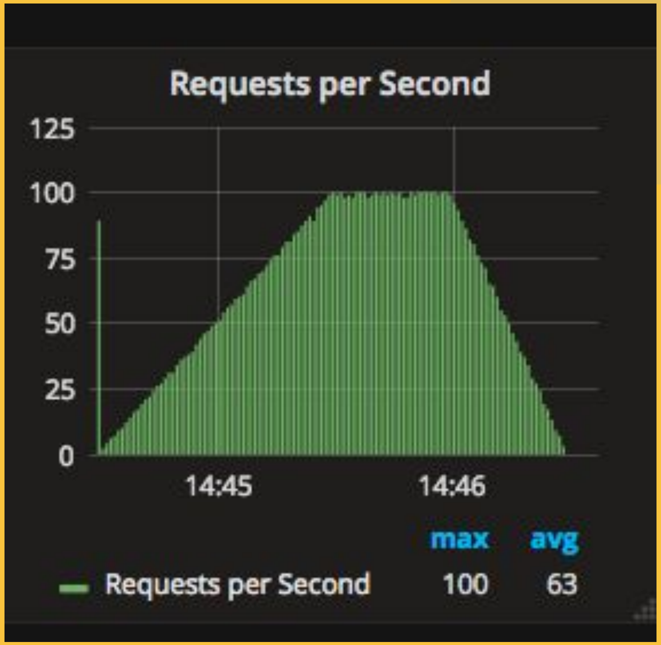
- can only increase

- number of requests
- number of errors
- ...

- compute rates

Counter

$$rate = \frac{N_{t1} - N_{t0}}{t_1 - t_0}$$

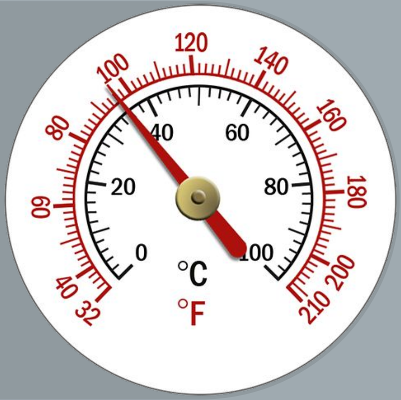


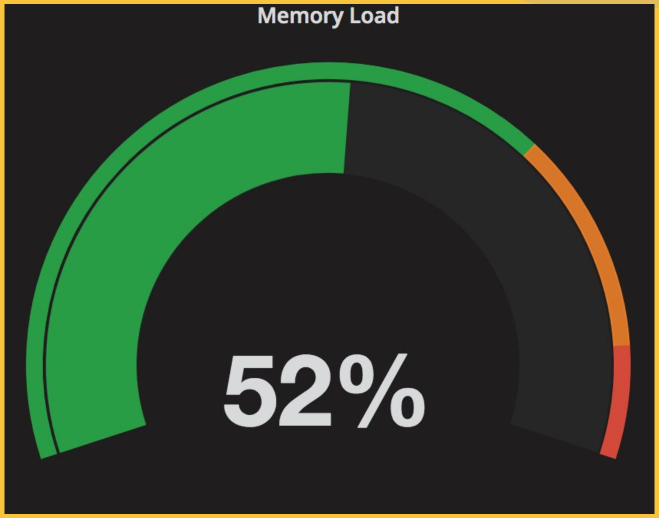
develeler

Gauge

- can increase and decrease

- temperature
- CPU load
- memory usage
- ...

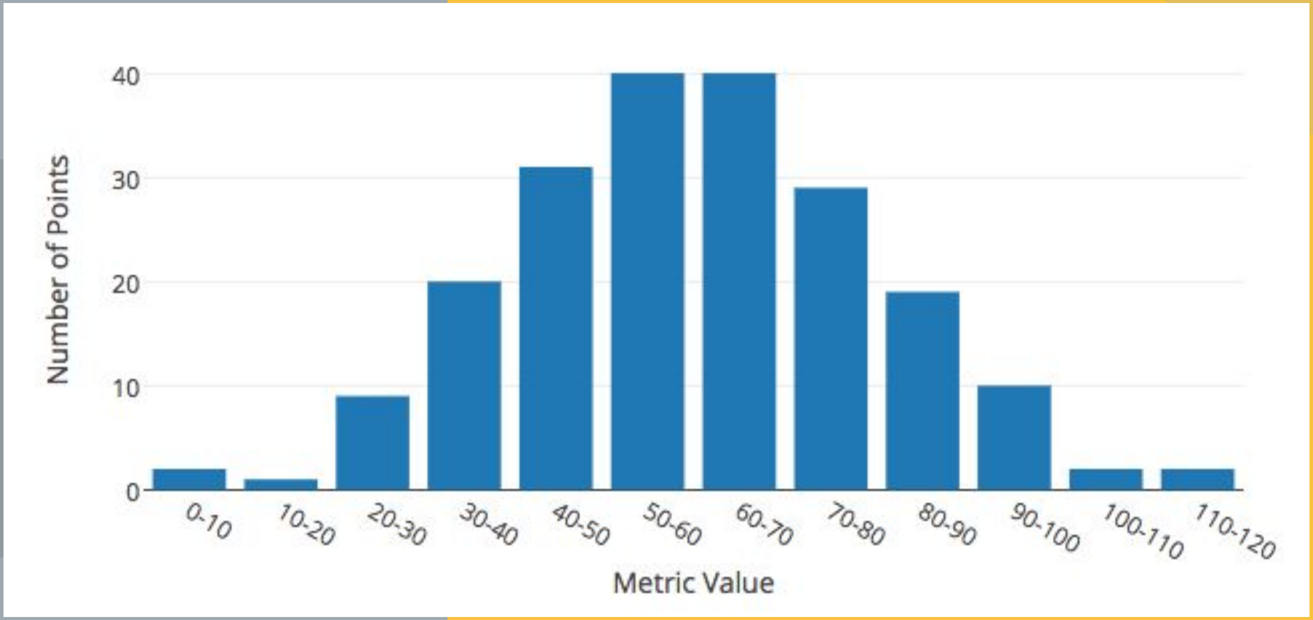




develer

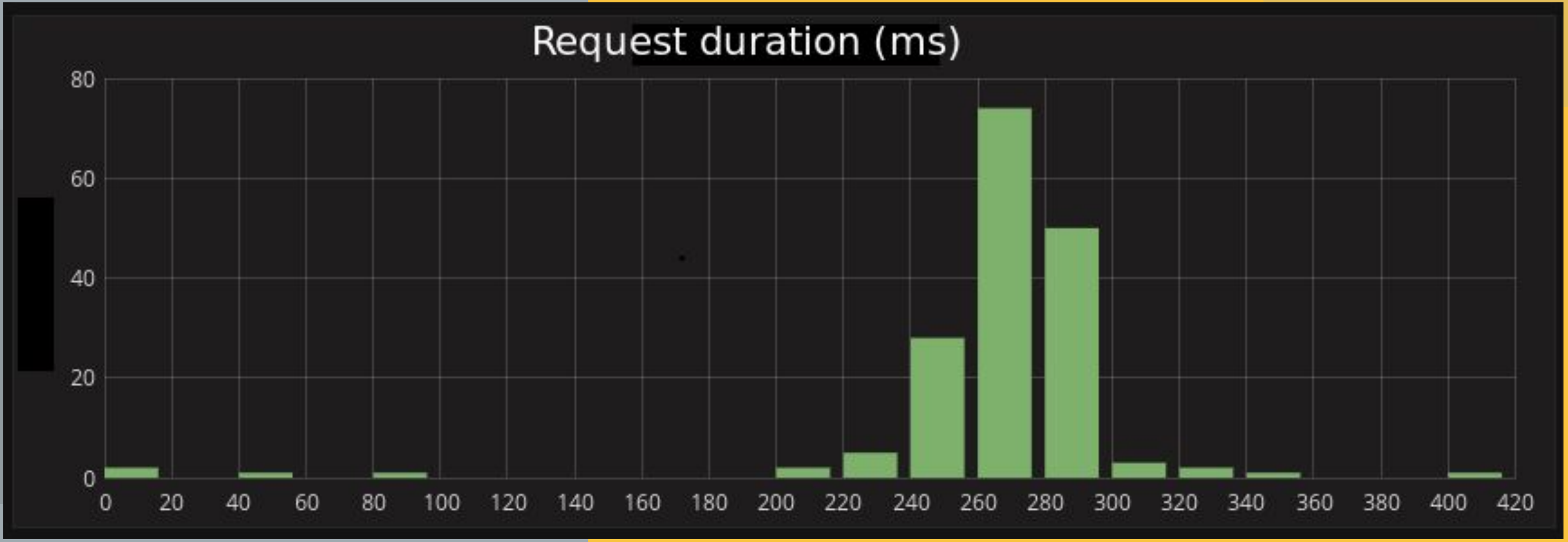
Histogram

- value distribution



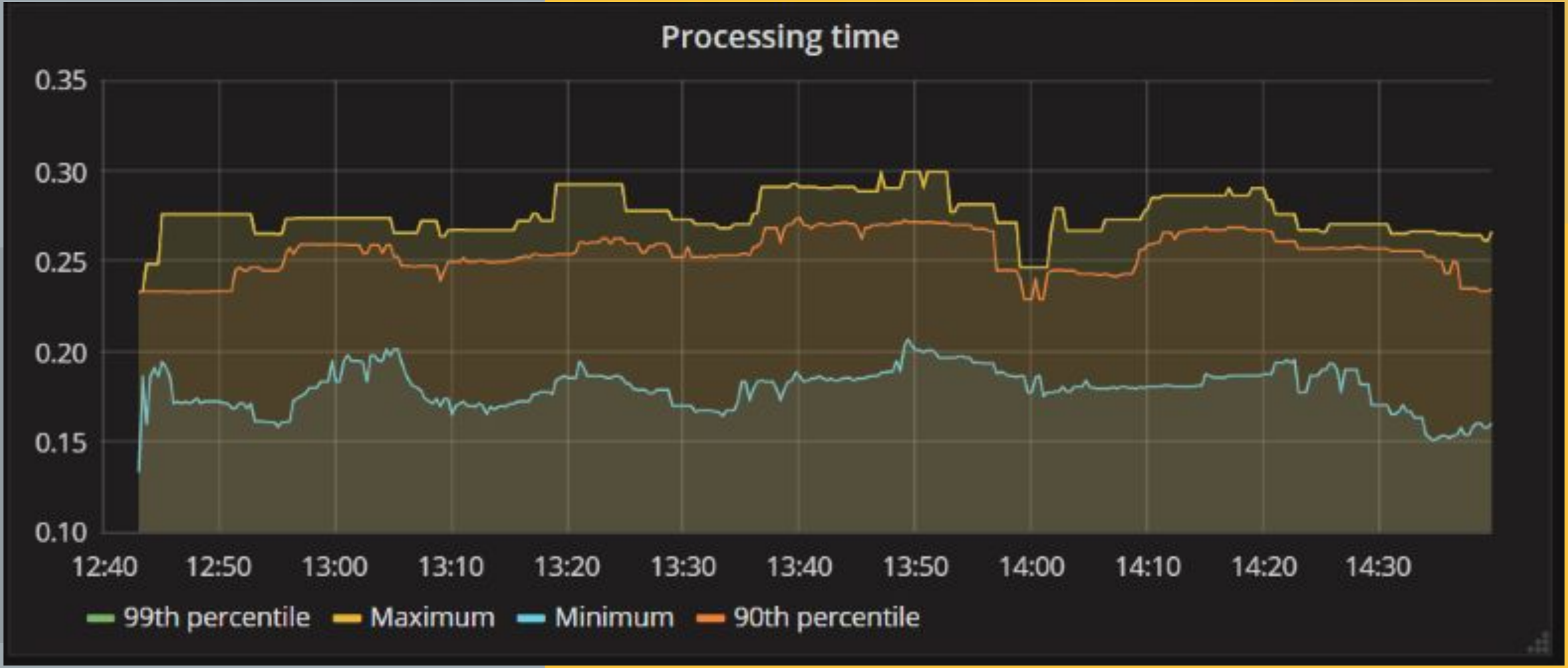
Histogram

- sample values
- counts them in bucket



Histogram

- median
- percentiles



develer

Hacking

```
$ git clone github.com/arl/monitoring
$ cd monitoring
$ cd cache
```

- **Add** a key-value pair to the cache:

  `http://localhost:8080/add?k=key&v=value`

- **Get** a cached value

  `http://localhost:8080/get?k=key`

- Count all requests
- Plot the number of requests per minute

TIPS:
use an http middleware
use rate (promQL)

Plot cache misses/hits

develer

- Record request duration
- Plot the median (50th percentile) over the last 10m

TIPS:
modify the previous middleware
use microseconds as unit
use histogram_quantile (promQL)

- Add a label named “endpoint” on request duration to have separate values for “add” and “get”

TIPS:
you can still do it with a middleware
use `promauto.NewHistogramVec`

Thank you!

Aurélien Rainone

develer

aurelien@develer.com

www.develer.com