SIG Instrumentation

Best practices for cluster observability through metrics and logging across all Kubernetes components

Optimizing Metric Rendering in *kube-state-metrics*

@mxinden

Max Inden

IndenML@gmail.com
@mxinden

Red Hat

Optimizing Metric Rendering in *kube-state-metrics*

@mxinden

Performance Optimizing with *Metrics* on *Kubernetes*

@mxinden

What is kube-state-metrics? Old version.

Exposes the state of a Kubernetes cluster in Prometheus metrics.

```
+-----------+                                                +-------------+
| APIServer |                                                | Prometheus  |
+-----------+                                                +-------------+
       |                                                                   |
       | New Pod p1                                                        |
       |---------------------->                                            |
       |                                                                   |
       |                                                                   |
       |                                                                   |
       |                                                                   |
       |                                                                   |
       |                                                                   |
       |                                                Get all metrics |
       |                           <----------------------------------------|
       |                                                                   |
       |                                                                   |
       |                                                                   |
       |                                                                   |
       |                                                                   |
       |                                                                   |
       |                                                                   |
       |                                                                   |
```

What is kube-state-metrics? Old version.

Exposes the state of a Kubernetes cluster in Prometheus metrics.

```
+-----------+      +--------------------+                        +-------------+
| APIServer |      | kubestatemetrics   |                        | Prometheus  |
+-----------+      +--------------------+                        +-------------+
      |                      |                                          |
      | New Pod p1           |                                          |
      |--------------------->|                                          |
      |          -----------\ |                                          |
      |          | Cache p1 |-|                                          |
      |          |----------| |                                          |
      |                      |                                          |
      |                      |                                          |
      |                      |                                          |
      |                      |                        Get all metrics |
      |                      |<----------------------------------------|
      |                      | --------------------------------\       |
      |                      |-| Render all Kubernetes objects |       |
      |                      | | into metrics                  |       |
      |                      | |-------------------------------|       |
      |                      |                                          |
      |                      | Metrics                                  |
      |                      |----------------------------------------->|
      |                      |                                          |
```

Kubernetes Object:

```yaml
- apiVersion: v1
  kind: Pod
  metadata:
    labels:
      app: kube-state-metrics
      pod-template-hash: 5fc64f676f
    name: kube-state-metrics-5fc64f676f-gl6v6
    namespace: monitoring
```

Prometheus Metric:

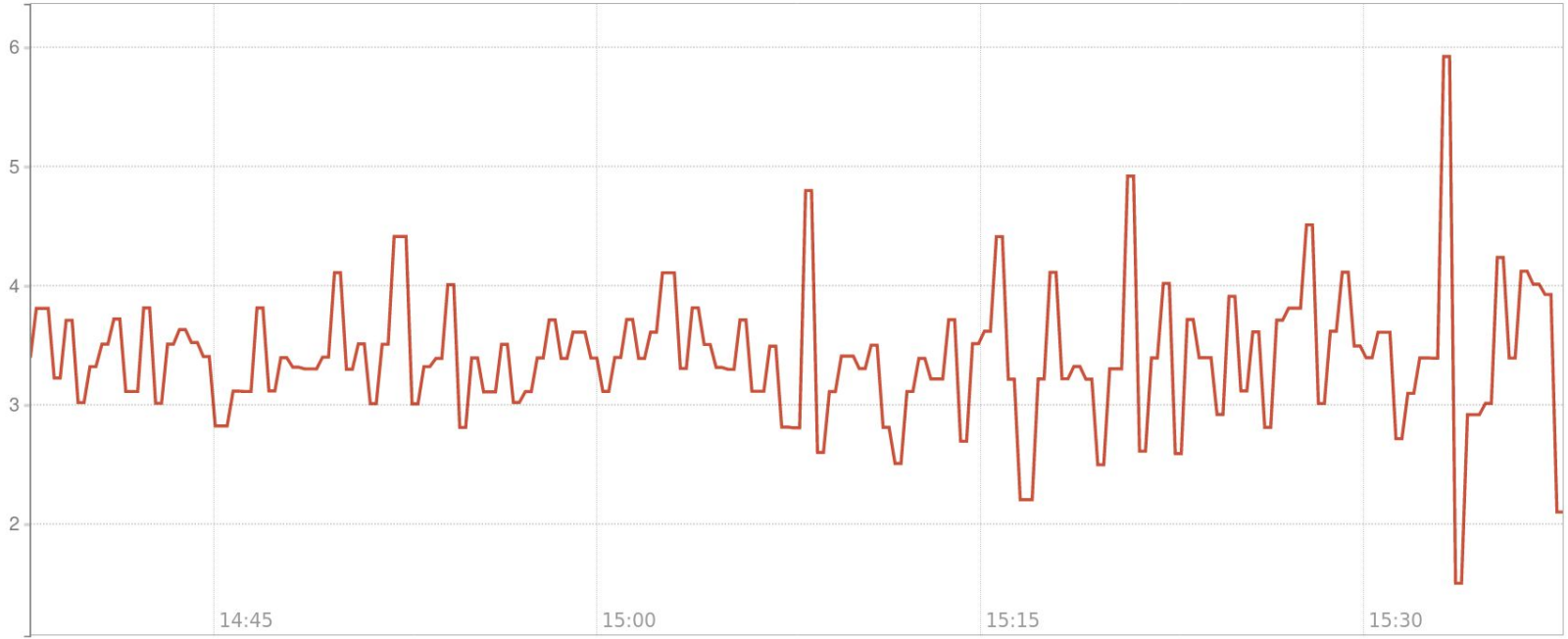
```
kube_pod_container_info{container="kube-state-metrics",namespace="monitoring",pod="kube-state-metrics-5fc64f676f-gl6v6"} 1

kube_pod_labels{label_app="kube-state-metrics",label_pod_template_hash="5fc64f676f",namespace="monitoring",pod="kube-state-metrics-5fc64f676f-gl6v6"} 1
```

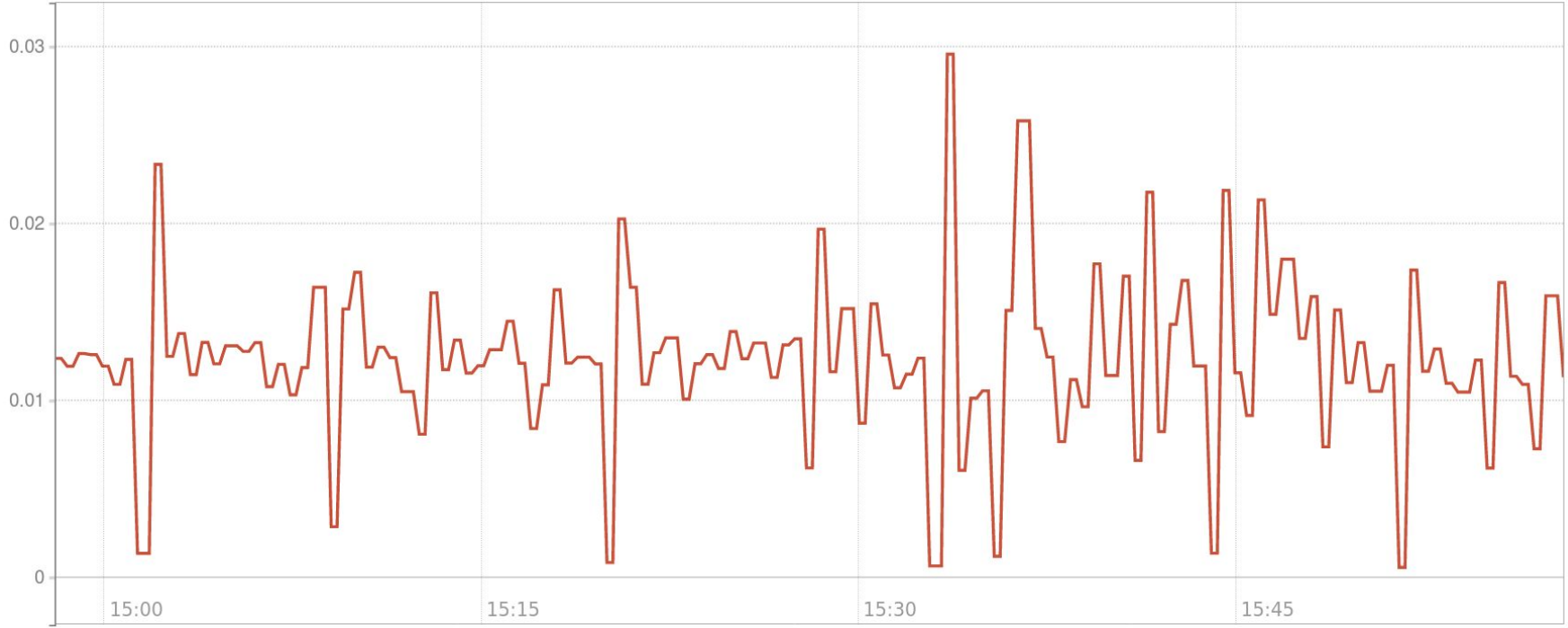
Problem

- High response times
  - 10s - 20s on big production clusters with ~50 mb of metrics
- High & unstable resource usage
  - Difficult to predict resource limits
  - ...

scrape_duration_seconds



container_cpu_usage_seconds



container_memory_usage_bytes

Caching

```
+-----------+       +--------------------+                        +-------------+
| APIServer |       | kubestatemetrics   |                        | Prometheus  |
+-----------+       +--------------------+                        +-------------+
     |                       |                                          |
     | New Pod p1            |                                          |
     |---------------------->|                                          |
     |         -----------\  |                                          |
     |         | Cache p1 |-|                                          |
     |         |----------| |                                          |
     |                       |                                          |
     |                       |                                          |
     |                       |                                          |
     |                       |                     Get all metrics      |
     |                       |<-----------------------------------------|
     |                       | --------------------------------\            |
     |                       |-| Render all Kubernetes objects |            |
     |                       | | into metrics                  |       |
     |                       | |-------------------------------|            |
     |                       |                                          |
     |                       | Metrics                                     |
     |                       |----------------------------------------> |
     |                       |                                          |
```

Caching

```
+-----------+           +--------------------+       +-------------+
| APIServer |           | kubestatemetrics   |       | Prometheus  |
+-----------+           +--------------------+       +-------------+
      |                           |                         |
      | New Pod p1                |                         |
      |-------------------------->|                         |
      |  ------------------------\ |                         |
      |  | Render metrics of p1 |-|                         |
      |  |----------------------| |                         |
      |-------------------------\ |                         |
      || Cache rendered metrics |-|                         |
      ||-------------------------| |                         |
      |                           |                         |
      |                           |                         |
      |                           |                         |
      |                           |        Get all metrics  |
      |                           |<------------------------|
      |                           |                         |
      |                           | Metrics                 |
      |                           |------------------------>|
      |                           |                         |
```

What to keep in cache?

Old: Kubernetes Object

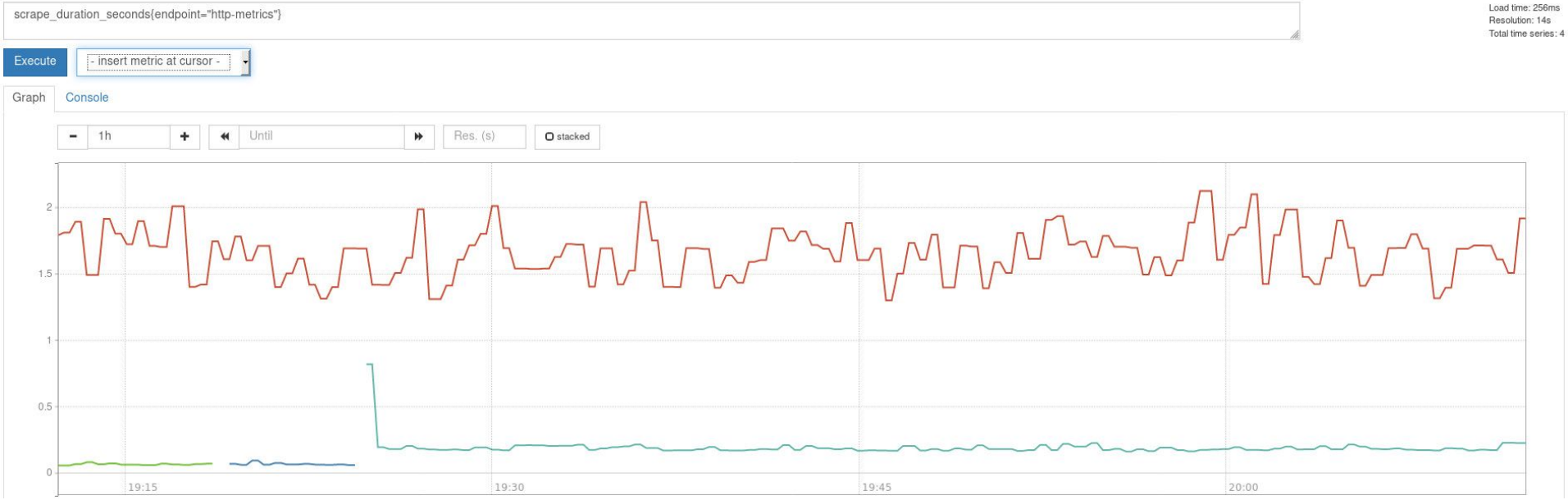
```yaml
- apiVersion: v1
  kind: Pod
  metadata:
    labels:
      app: kube-state-metrics
      pod-template-hash: 5fc64f676f
    name: kube-state-metrics-5fc64f676f-gl6v6
    namespace: monitoring
```

New: Prometheus Metric

```
kube_pod_container_info{container="kube-state-metrics",namespace="monitoring",pod="kube-state-metrics-5fc64f676f-gl6v6"} 1

kube_pod_labels{label_app="kube-state-metrics",label_pod_template_hash="5fc64f676f",namespace="monitoring",pod="kube-state-metrics-5fc64f676f-gl6v6"} 1
```

Performance improvement through caching



Use just Reflector not whole Informers

```go
// Reflector watches a specified resource
// and causes all changes to be reflected
// in the given store.
type Reflector struct {
    // [...]

    // The type of object we expect
    // to place in the store.
    expectedType reflect.Type
    // The destination to sync up
    // with the watch source
    store Store
    // listerWatcher is used to perform
    // lists and watches.
    listerWatcher ListerWatcher

    // [...]
}
```

```go
type Store interface {
    Add(obj interface{}) error
    Update(obj interface{}) error
    Delete(obj interface{}) error
    List() []interface{}
    ListKeys() []string
    Get(obj interface{}) (item interface{}, exists bool, err error)
    GetByKey(key string) (item interface{}, exists bool, err error)

    // [...]
}
```

Compression

```
# HELP kube_secret_type Type about secret.
# TYPE kube_secret_type gauge
kube_secret_type{namespace="default",secret="test-0",type="Opaque"} 1
kube_secret_type{namespace="default",secret="test-1",type="Opaque"} 1
kube_secret_type{namespace="default",secret="test-2",type="Opaque"} 1
```

https://golang.org/pkg/compress/gzip/

https://github.com/NYTimes/gziphandler

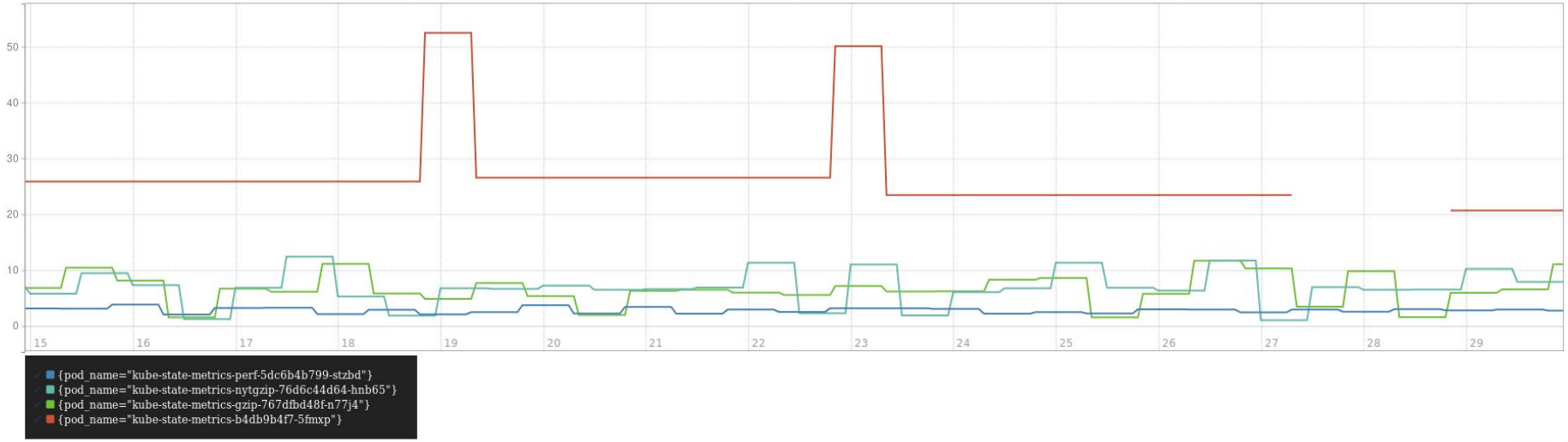
```
sum((irate(container_cpu_usage_seconds_total{container_name="kube-state-metrics",namespace="mxinden"}[5m]))*100) by (pod_name)
```

Load time: 242ms
Resolution: 3s
Total time series: 4

Execute | - insert metric at cursor -

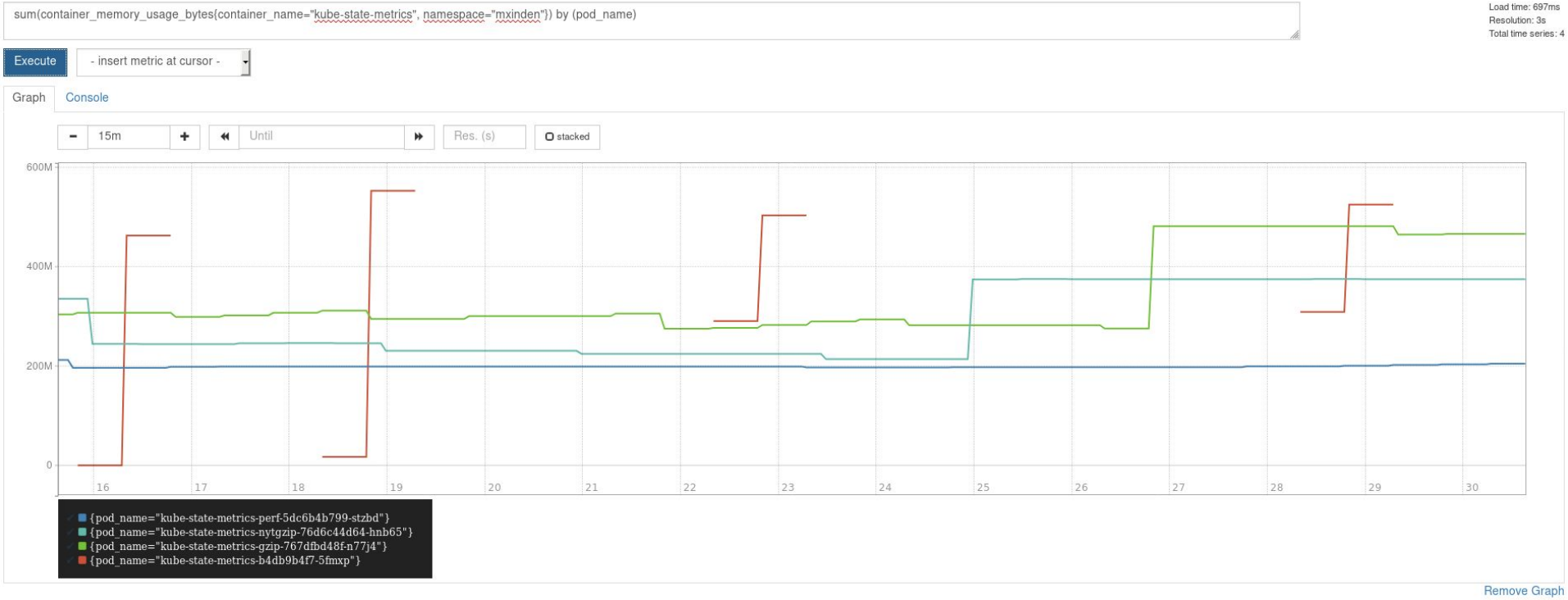
Graph | Console

15m | Until | Res. (s) | stacked



Remove Graph

metrics + gzip(metrics) > metrics



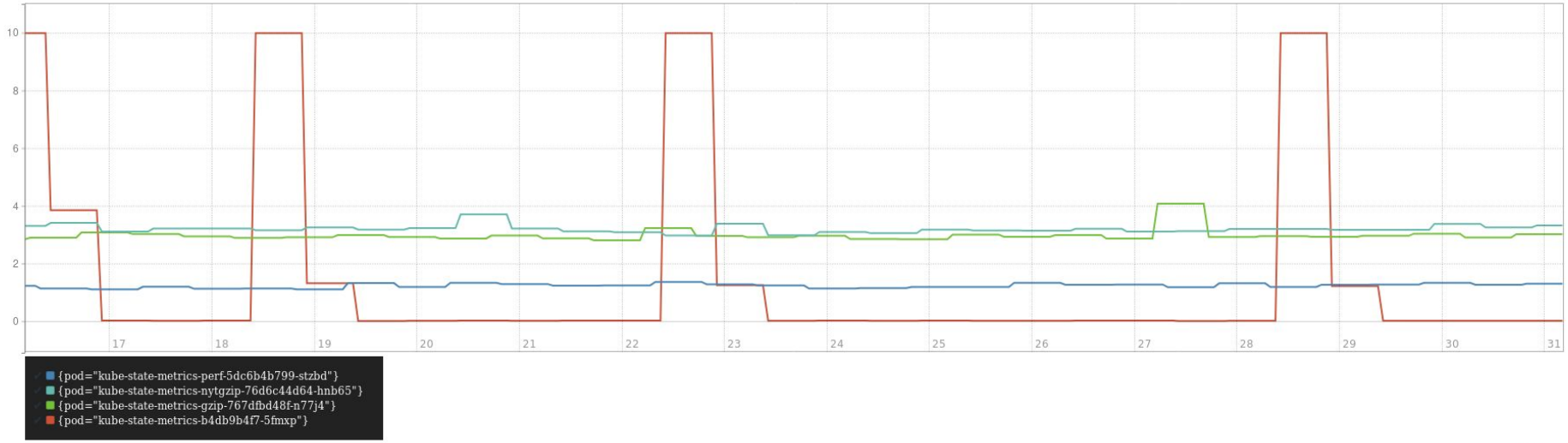
```
sum(scrape_duration_seconds{endpoint="http-metrics"}) by (pod)
```

Load time: 823ms
Resolution: 3s
Total time series: 4

Execute | - insert metric at cursor -

Graph | Console

15m | Until | Res. (s) | stacked



Remove Graph

Compression

```
size(metrics + gzip(metrics)) > size(metrics)

Improved network throughput < Lost CPU cycles compressing

Higher CPU utilization
```

Hard code float-to-string common cases

```go
func writeFloat(w *strings.Builder, f float64) {
    switch {
    case f == 1:
        w.WriteByte('1')
    case f == 0:
        w.WriteByte('0')
    case f == -1:
        w.WriteString("-1")
    case math.IsNaN(f):
        w.WriteString("NaN")
    case math.IsInf(f, +1):
        w.WriteString("+Inf")
    case math.IsInf(f, -1):
        w.WriteString("-Inf")
    default:
        // [...]
    }
}
```

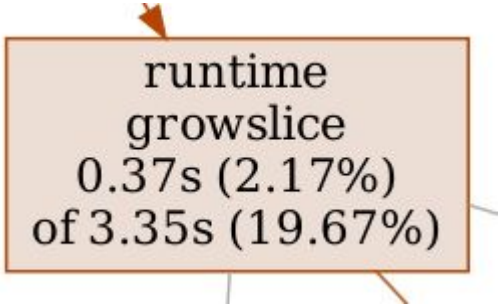
Golang strings.Builder

```go
// A Builder is used to efficiently build
// a string using Write methods.
// It minimizes memory copying.
type Builder struct {
    // [...]
    buf  []byte
}
```

Version	Scrape duration	CPU usage	RSS peak	% in GC
v1.4	22s	1.0	2.5GiB	0.015
newprom_gzip	16s	0.8	2.2 GiB	0.010
newprom_nogzip	13s	0.6	2.4 GiB	0.014
mxinden_gzip	11s	0.3	0.81 GiB	0.006
mxinden_nogzip	7s	0.16	0.73 GiB	0.004

v1.5.0

Memory pre-allocation



```go
// growslice handles slice growth during append.
// It is passed the slice element type, the old
// slice, and the desired new minimum capacity,
// and it returns a new slice with at least that
// capacity, with the old data copied into it.
// [...]
func growslice(et *_type, old slice, cap int) slice {
```

```go
{
    Name: "kube_pod_container_info",
    // [...]
    GenerateFunc: wrapPodFunc(func(p *v1.Pod) *metric.Family {
        ms := []*metric.Metric{}
        // [...]

        for _, cs := range p.Status.ContainerStatuses {
            ms = append(ms, &metric.Metric{
                // [...]
            })
        }

        // [...]
    }),
},
```

```go
{
    Name: "kube_pod_container_info",
    // [...]
    GenerateFunc: wrapPodFunc(func(p *v1.Pod) *metric.Family {
        ms := make([]*metric.Metric, len(p.Status.ContainerStatuses))
        // [...]

        for i, cs := range p.Status.ContainerStatuses {
            ms[i] = &metric.Metric{
            // [...]
            }
        }

        // [...]
    }),
},
```

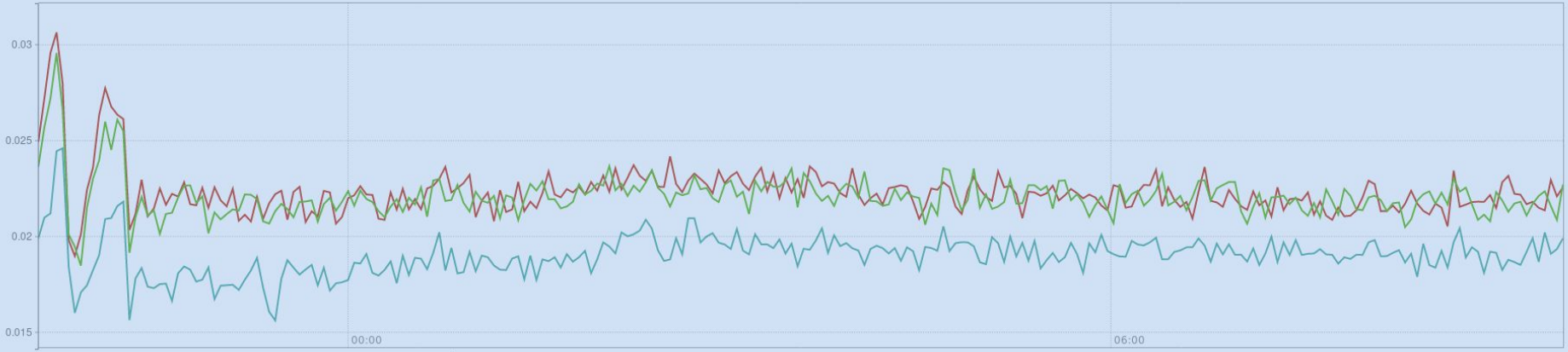
sum(rate(container_cpu_usage_seconds_total{container_name="kube-state-metrics", pod_name=~".*v1-6-0-rc-0-.*"}[5m])) by (pod_name)

Load time: 312ms
Resolution: 172s
Total time series: 3

Execute | - insert metric at cursor -

Graph | Console

12h | Until | Res. (s) | stacked



{pod_name="kube-state-metrics-v1-6-0-rc-0-5-7cf69f8d48-5wn74"}
{pod_name="kube-state-metrics-v1-6-0-rc-0-4-758fb8d5ff-jg9ts"}
{pod_name="kube-state-metrics-v1-6-0-rc-0-0-756b46798c-zhgft"}

Remove Graph

```go
func growslice(et *_type, old slice, cap int) slice {
  // [...]

    newcap := old.cap
    doublecap := newcap + newcap

    if cap > doublecap {
        newcap = cap
    } else {
        if old.len < 1024 {
            newcap = doublecap
        } else {
          // [...]
        }
    }
}
```

[]byte != string

- Strings are immutable

```
+-----------+     +-----------+                +-------------+         +-------------+
| APIServer |     | Collector |           | MetricStore |         | Prometheus  |
+-----------+     +-----------+                +-------------+         +-------------+
      |                 |                           |                       |
      | New Pod p1      |                           |                       |
      |---------------->|                           |                       |
      |                 |                           |                       |
      |                 | p1 metrics (string)       |                       |
      |                 |---------------------------->|                     |
      |                 |                           |                       |
      |                 |                           |                       |
      |                 |                           |                       |
      |                 |                           |       Get all metrics |
      |                 |                           |<----------------------|
      |                 |                           |                       |
      |                 |                           | metrics ([]byte)      |
      |                 |                           |---------------------->|
      |                 |                           |                       |
```

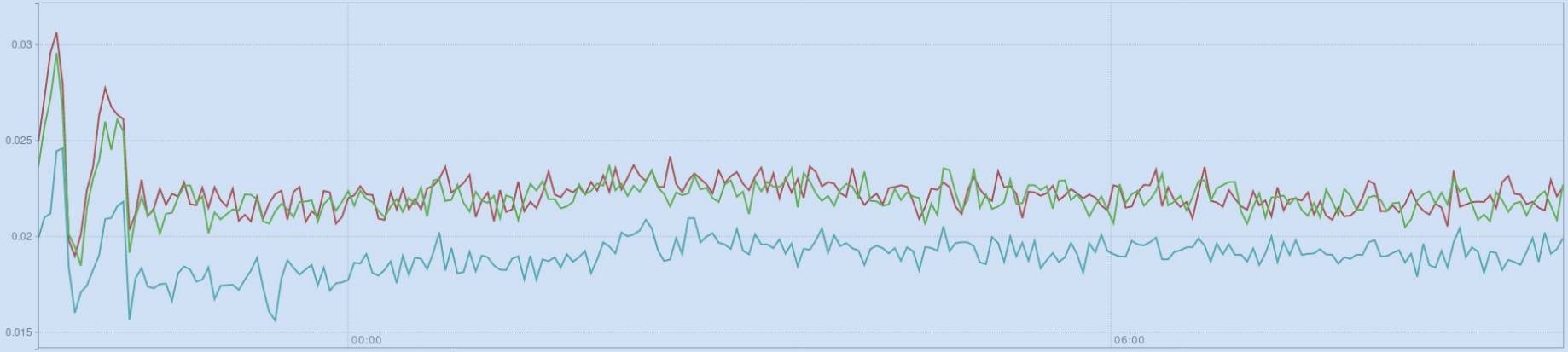
sum(rate(container_cpu_usage_seconds_total{container_name="kube-state-metrics", pod_name=~".*v1-6-0-rc-0-.*"}[5m])) by (pod_name)

Load time: 312ms
Resolution: 172s
Total time series: 3

Execute | - insert metric at cursor -

Graph | Console

12h | Until | Res. (s) | stacked



{pod_name="kube-state-metrics-v1-6-0-rc-0-5-7cf69f8d48-5wn74"}
{pod_name="kube-state-metrics-v1-6-0-rc-0-4-758fb8d5ff-jg9ts"}
{pod_name="kube-state-metrics-v1-6-0-rc-0-0-756b46798c-zhgft"}

Remove Graph

v1.6.0

Future

- Introducing sharding based on object id instead of Kubernetes type
- Reduce heap escapes
- Optimize memory alignment on structs in hot path
- …
- (Commutative compression)

Take aways

- Monitor your cluster and application!
- Use **Monitoring** for alerting **and performance analysis**

Golang memory model documentation:

> If you must read the rest of this document to understand the behavior of your program, you are being too clever.

> Don't be clever.