Great Cardinality Disasters of Our Time

Bryan Boreham (@bboreham)
Chris Marchbanks (@csmarchbanks)

weaveworks
splunk>

Who are we?

Bryan Boreham

Director of Engineering, Weaveworks

Maintainer of Cortex and CNI

Part-time SRE on Weave Cloud - GitOps and Prometheus-as-a-Service

Chris Marchbanks

Senior Software Engineer, Splunk

Maintainer of Cortex and Prometheus

Building internal observability platform for Splunk Cloud

Why are we talking about “Cardinality”?

Prometheus blows up.

> **CAUTION:** Remember that every unique combination of key-value label pairs represents a new time series, which can dramatically increase the amount of data stored. Do not use labels to store dimensions with high cardinality (many different label values), such as user IDs, email addresses, or other unbounded sets of values.

Every metrics system runs into issues with Cardinality

How much is too much?

Avoid querying more than ~100,000 series at a time.
E.g. an instant query like count(metric_a):

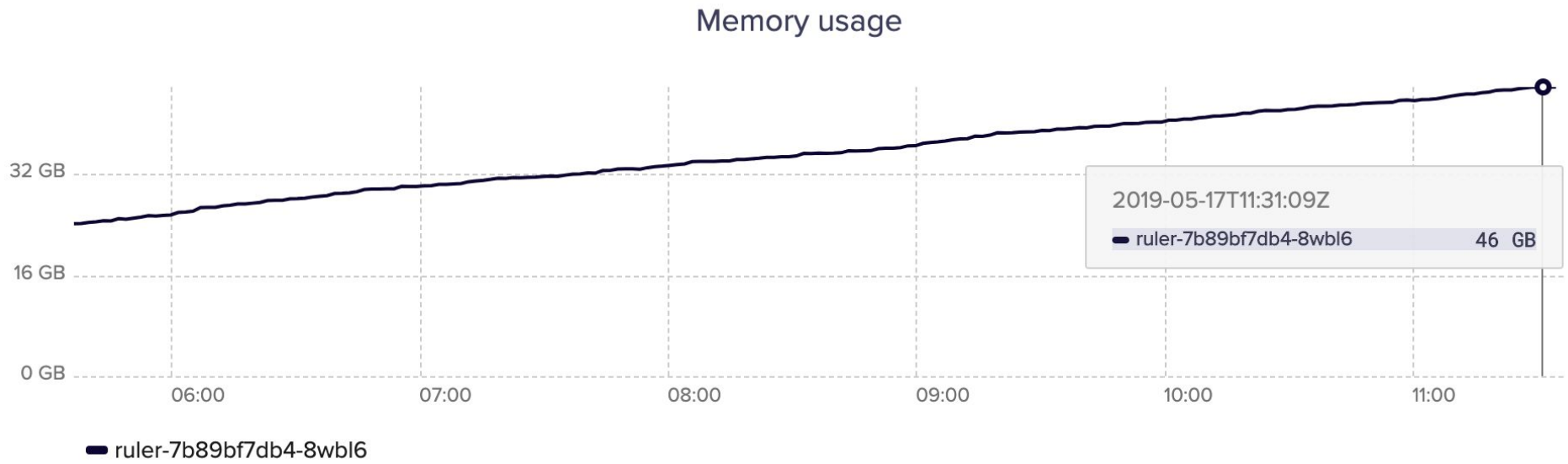
- 100,000 series in ~1.5 seconds
- 200,000 series in ~5 seconds

A single Prometheus can handle >10,000,000 series in memory.

- Startup can take 15+ minutes
- Heap can exceed 100 GB

Disaster #1: Alerts

Disaster #1: First thing we knew



Disaster #1: The explanation

User had a rule configured:

```
alert: InvalidTagSequenceTotal
expr:  invalidtag_sequence_total > 0
for:   1m
```

`count(invalidtag_sequence_total>0)` was over 40,000.

Prometheus creates two time-series per alert, so it was trying to update 80,000 series every 15 seconds.

Total of 300,000 alerts over the whole period.

Disaster #1: What to learn

Careful with the query in alerts

```
expr:  count(invalidtag_sequence_total > 0) > 0
```

Or perhaps:

```
expr:  topk(10, invalidtag_sequence_total > 0)
```

Disaster #2: 100KB label values

KubeCon | CloudNativeCon
North America 2019

A bug in Istio wrote raw error messages to the err label, causing Prometheus label values to be 100KB unique values.

Example label value

Error adding/updating cluster(s) outbound|50052||clientservice.clientnamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||clientservice.clientnamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||clientservice.clientnamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||clientservice.clientnamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||clientservice.clientnamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|8443||aservice.somenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50051||anotherservice.anotherservicenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|9091||service3.yetanotherservicenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50051||service4.yetanotherservicenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|9000||service4.yetanotherservicenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||servicename.servicenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||servicename.servicenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||servicename.servicenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||servicename.servicenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||servicename.servicenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||servicename.servicenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||servicename.servicenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||servicename.servicenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||servicename.servicenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||servicename.servicenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||servicename.servicenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||servicename.servicenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||servicename.servicenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||servicename.servicenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||servicename.servicenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||servicename.servicenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||servicename.servicenamespace.svc.cluster.local: Invalid path: /etc/certs/cert-chain.pem, outbound|50052||servicename.servicenamespace.svc.cluster.local: Invalid path:

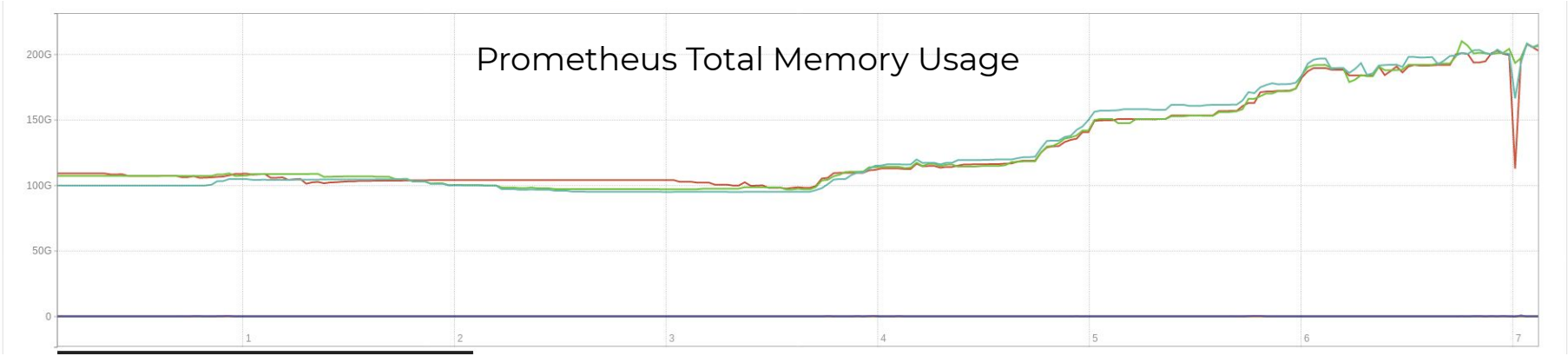
And continuing for many more lines...

Disaster #2: Full Prometheus Outage

Prometheus instances all OOMed simultaneously.

Growth of the symbol table in Prometheus:
`prometheus_tsdb_symbol_table_size_bytes`

Symbol table size grew from 5 GB to 60 GB.



Disaster #2: What to learn

Do not put raw messages into a label value!

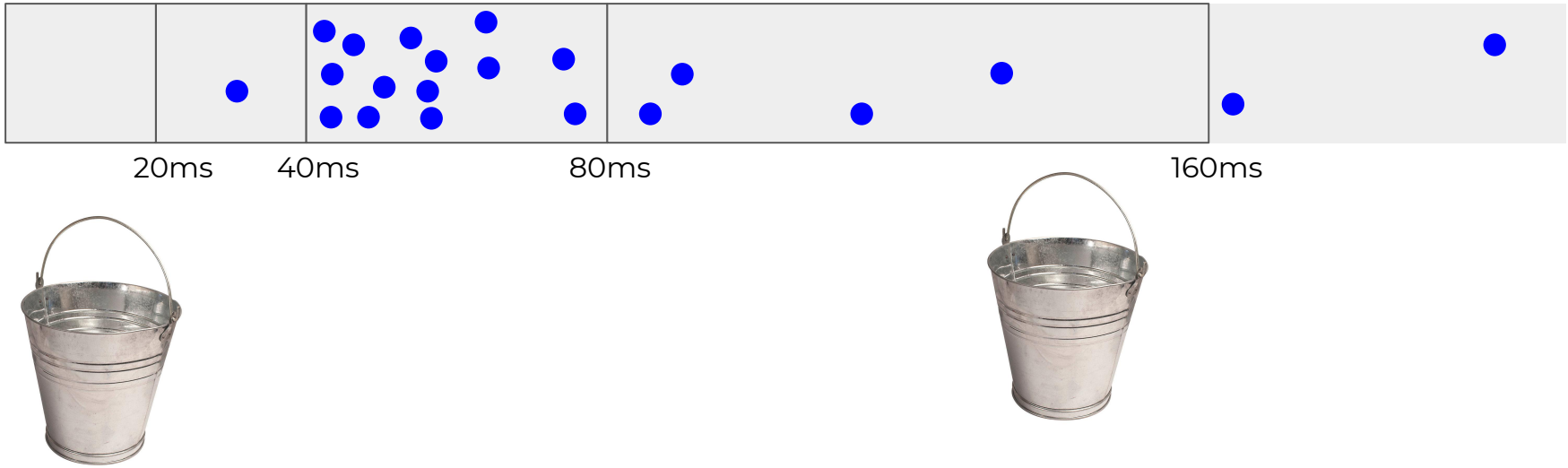
Prefer a fixed size enum, and logging the raw messages.

Until the source can be fixed, use a relabeling rule to drop the offending metric.

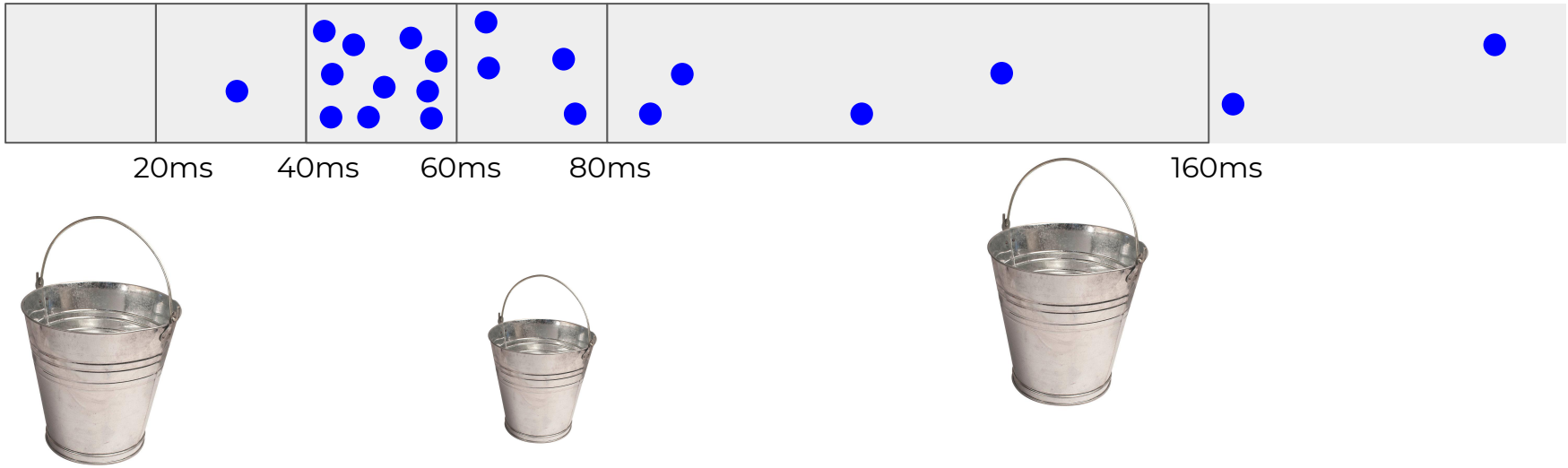
Even well-used projects can have bugs in them.
Thank you to the Istio team for quickly fixing our issue.

Disaster #3: Buckets

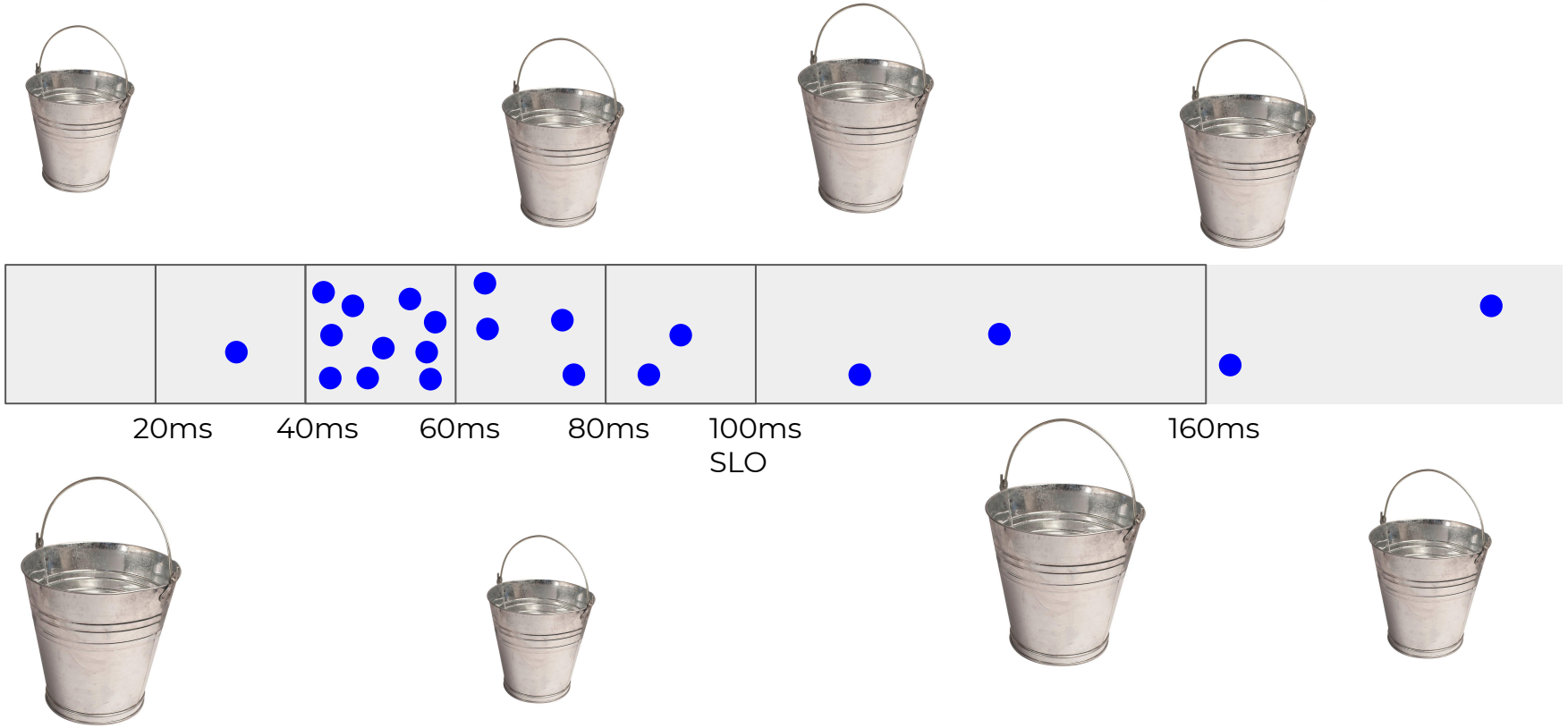
A Prometheus Histogram counts the number of samples in each bucket.



Disaster #3: Buckets



Disaster #3: Buckets




ID 70136132 © | Dreamstime.com

Disaster #3: kube-dns histogram

github.com/kubernetes/dns/pkg/sidecar/dnsprobe.go

```
prometheus.HistogramOpts{
    Name:       "kubedns_latency_ms",
    Buckets:    prometheus.LinearBuckets(0, 10, 500),
}
```

Five. Hundred. Buckets.

Bryan fixed it for you.

pkg/sidecar/dnsprobe.go

```
@@ -95,7 +95,7 @@ func (p *dnsProbe) registerMetrics(options *Options) {
                Subsystem: dnsProbeSubsystem,
                Name:      p.Label + "_latency_ms",
                Help:      "Latency of the DNS probe request " + p.Label,
-               Buckets:   prometheus.LinearBuckets(0, 10, 500),
+               Buckets:   prometheus.ExponentialBuckets(0.25, 2, 16), // from 0.25ms to 8 seconds
        })
        prometheus.MustRegister(p.latencyHistogram)
```

Disaster #4: All the metric names!

A team added Prometheus metrics using a library that didn't expose tags. What can you do? Try putting them in names?

```
my_service_http_<route>_<tenant>_<status_code>_total
```

Queries like:

```
sum({__name__=~"my_service_http_getInfo_.*_total"})
```

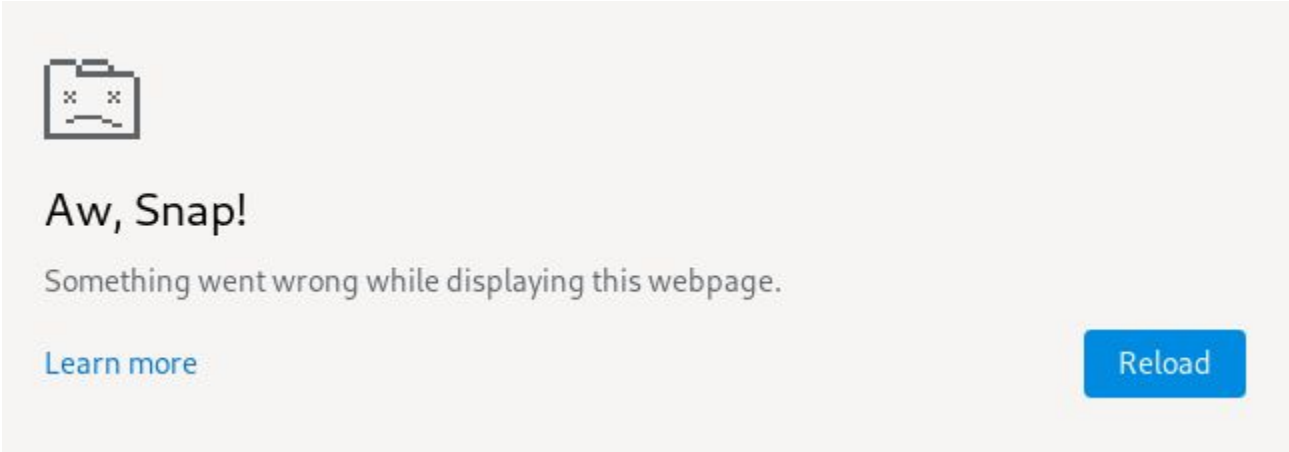
Do not actually do this...

Disaster #4: Sad browsers

Prometheus /graph UI becomes unusable

Prometheus (and Grafana) pulls back all metric names for autocomplete

Some improvements have been made, but still work to be done



How many metric names?!

[redacted] commented on Sep 3

```
curl http://prometheus:9090/api/v1/label/__name__/values > metricnames.json
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  121M    0  121M    0     0  1191k      0 --:--:--  0:01:44 --:--:-- 1551k

cat metricnames.json | tr '"' ' ' | tr -d "," | wc -w
2129511
```

We have about 2 million metric names.

Disaster #4: What to learn

Do not put labels into a metric name.

The overall cardinality will be the same, but leads to many inefficiencies.

If needed, you can use relabelling rules to parse labels out of names - but it is better to fix them at the source.

How to troubleshoot?

Extreme cardinality can be unpleasant to deal with.

What are the techniques to apply when you think it is happening?

```
count by(__name__)({__name__=~".+"})
```

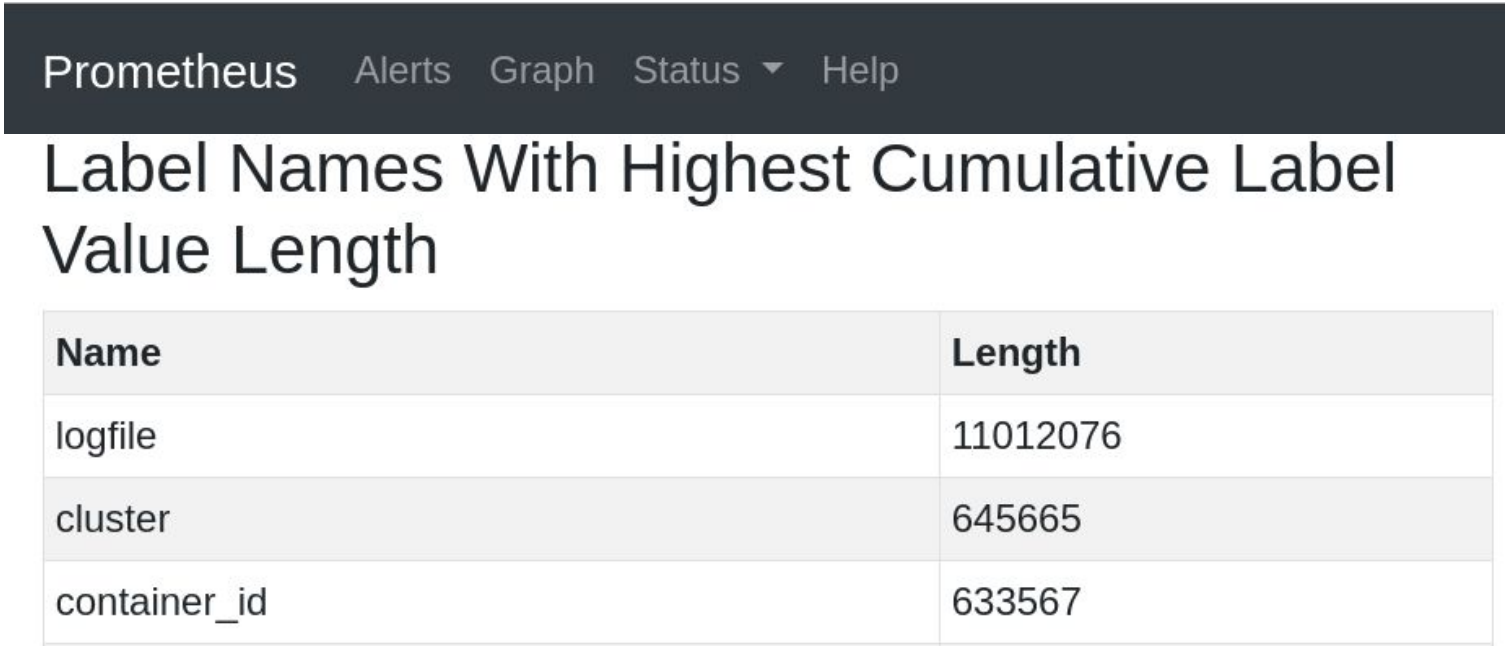
How to troubleshoot?

In larger instances, `__name__` queries stop working. Instead,

1. How many series are in memory?
   - `prometheus_tsdb_head_series`
2. How much space do my unique strings take?
   - `prometheus_tsdb_symbol_table_size_bytes`
3. How many samples am I scraping by job?
   - `sum(scrape_samples_scraped) by (job)`
4. In 2.14.0 use the status page! http://prometheus:9090/status

Troubleshooting example

My symbol table size was greater than 1 GB in an instance.



What series have logfile?

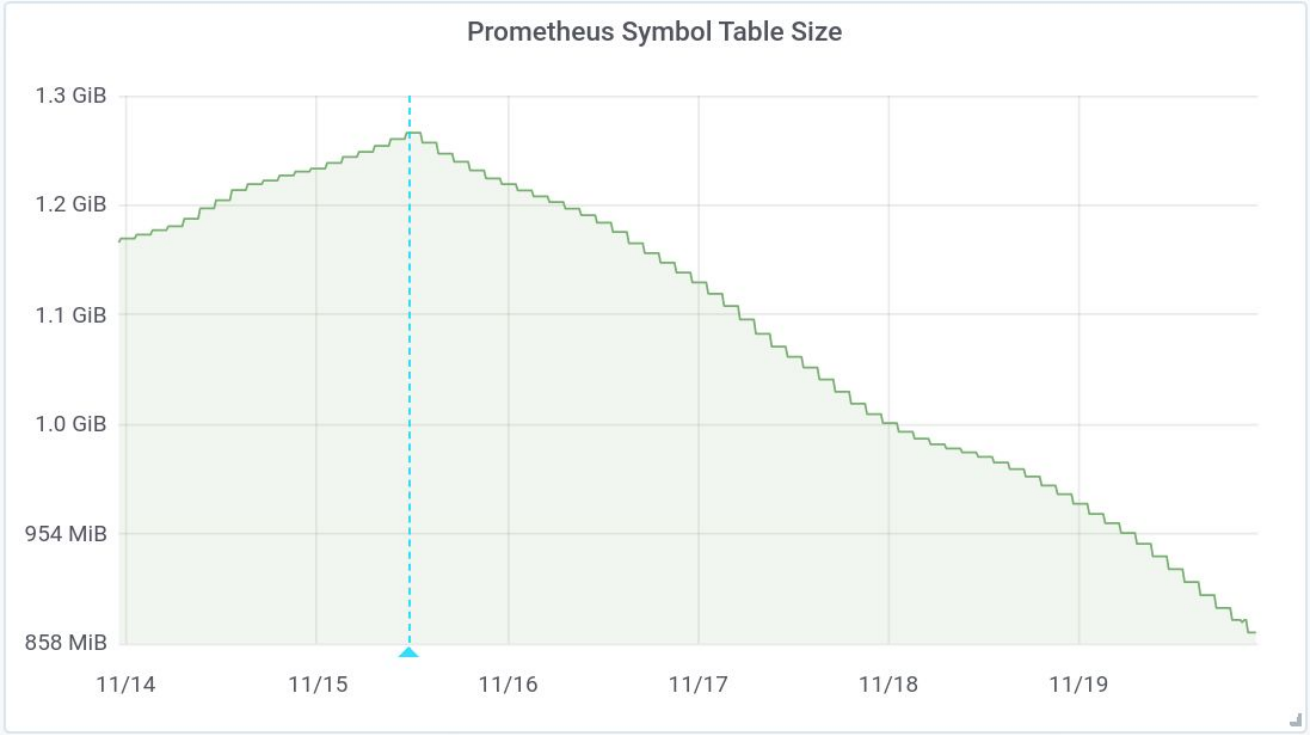
Query: **`count({logfile!=""}) by (job, __name__)`**

Result:

Element	Value
mtail_log_rotations_total {job="mtail"}	783
mtail_log_lines_total {job="mtail"}	48407

Results

Reduce the number of containers mtail is following.



What cardinality can you find?


KubeCon
CloudNativeCon
North America 2019
Questions?
Bryan Boreham (@bboreham)
Chris Marchbanks (@csmarchbanks)
weaveworks
splunk>