

How we used Jaeger and Prometheus to Deliver Lightning-Fast User Queries

Bryan Boreham

Director of Engineering, Weaveworks



Who knows...

Jaeger?

Prometheus?

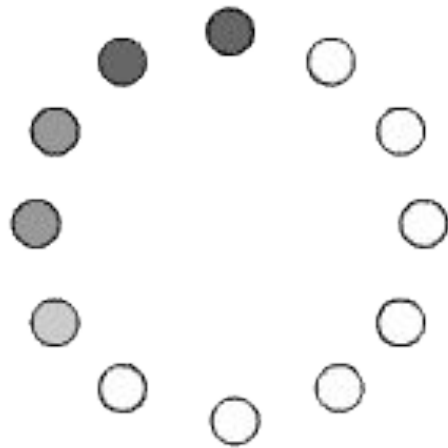
Weaveworks?



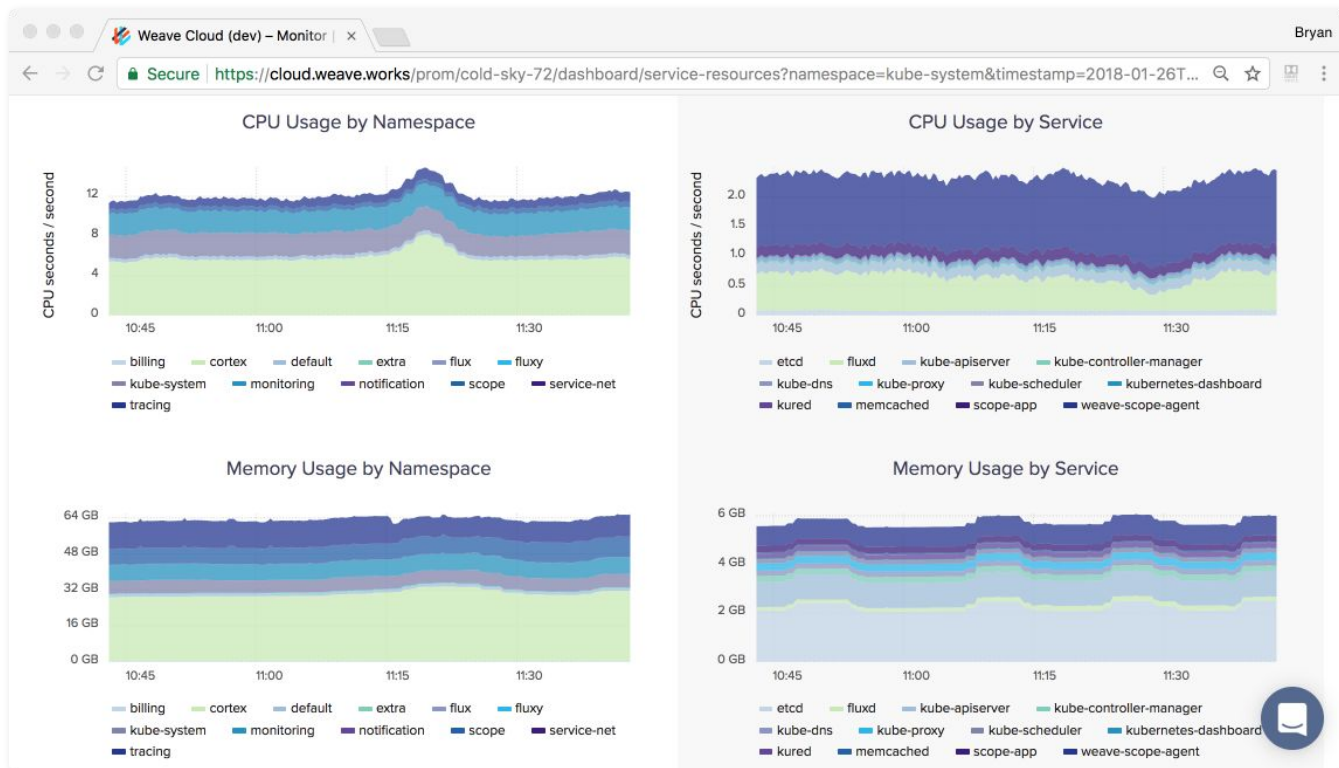
Today's story...

- What are Prometheus and Jaeger
- What to look for
- Applying the information

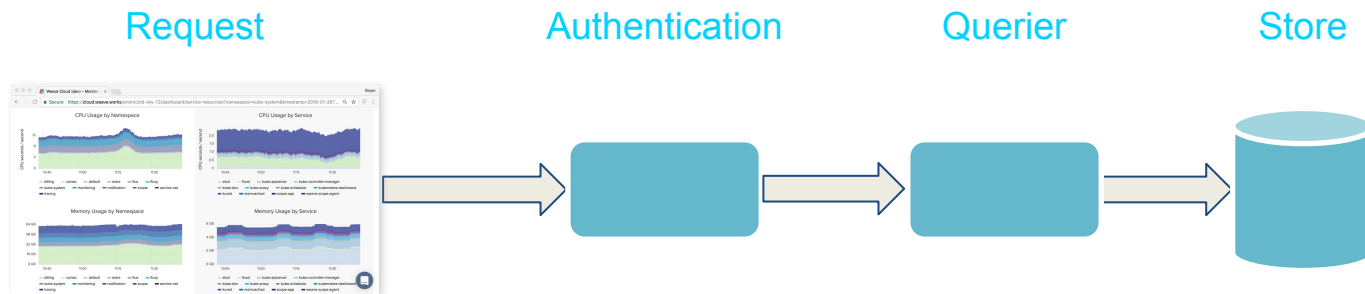




I wanted this!



What am I talking about?



Weave Cortex- multi-tenant Prometheus
<https://github.com/weaveworks/cortex>

The three most important things in software optimisation



Measure,



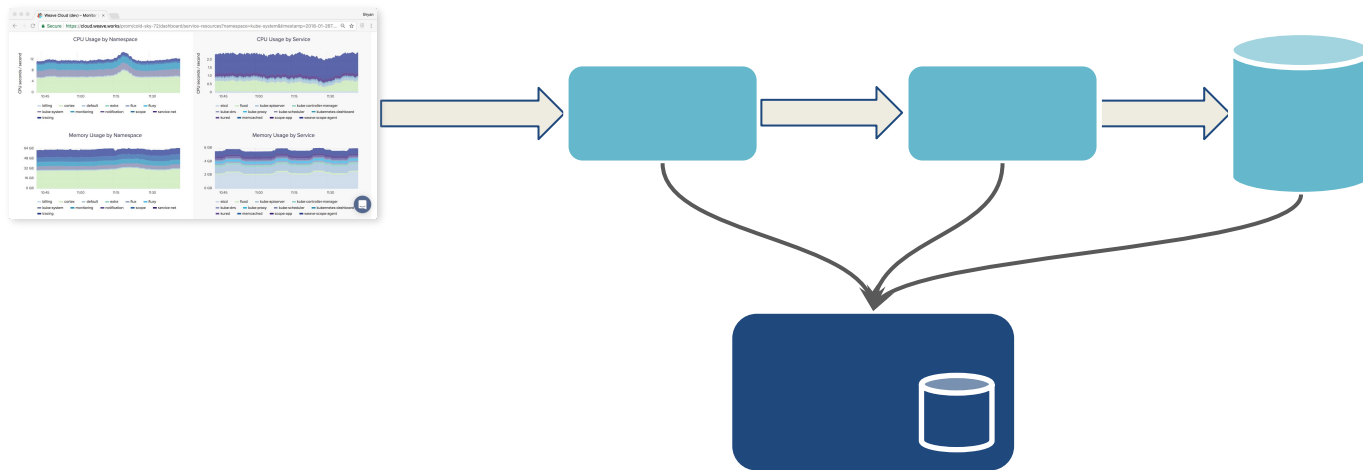
Measure,



Measure.



Measuring a distributed system



Prometheus: time-series metrics



Basic metrics:

- CPU, Memory, Disk, Network

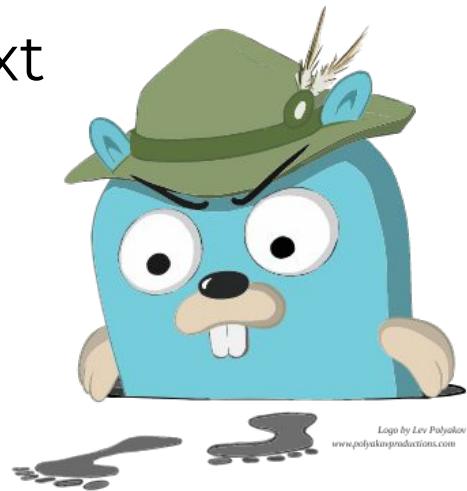
Service metrics:

- Request rate
- Error rate
- Delay (Latency)



Jaeger: Getting inside a request

- What is my code really doing?
- What is the pattern between operations?
- Trace from one component to the next

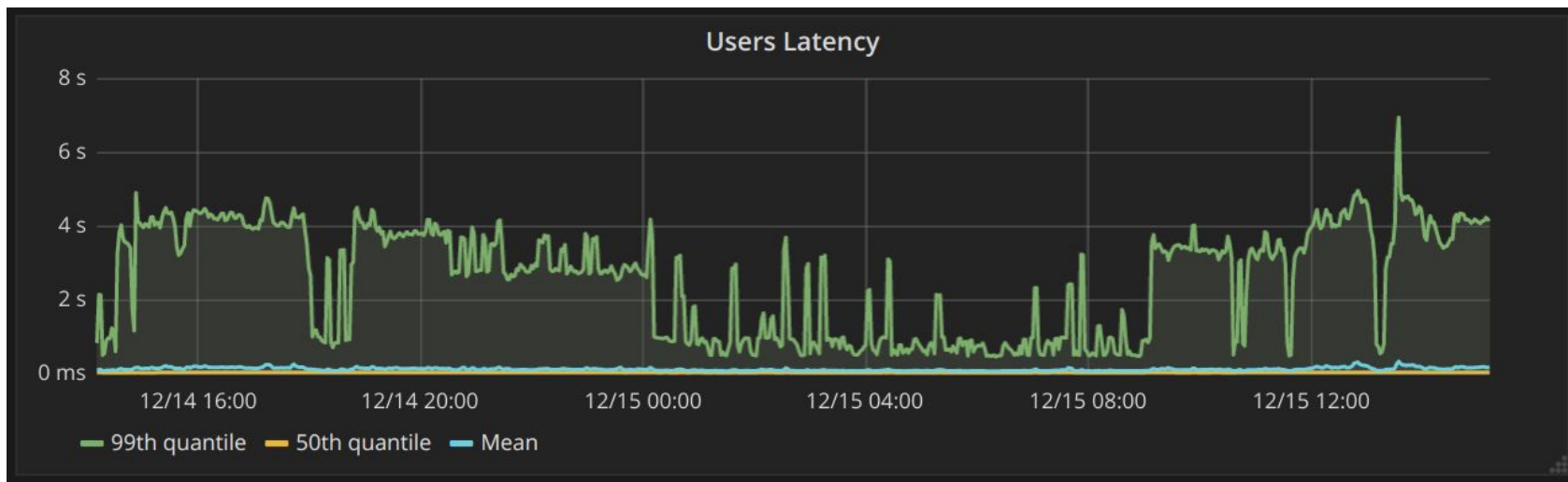


Logo by Lev Polyakov
www.polyakovproductions.com

The case of the mysterious user service latency



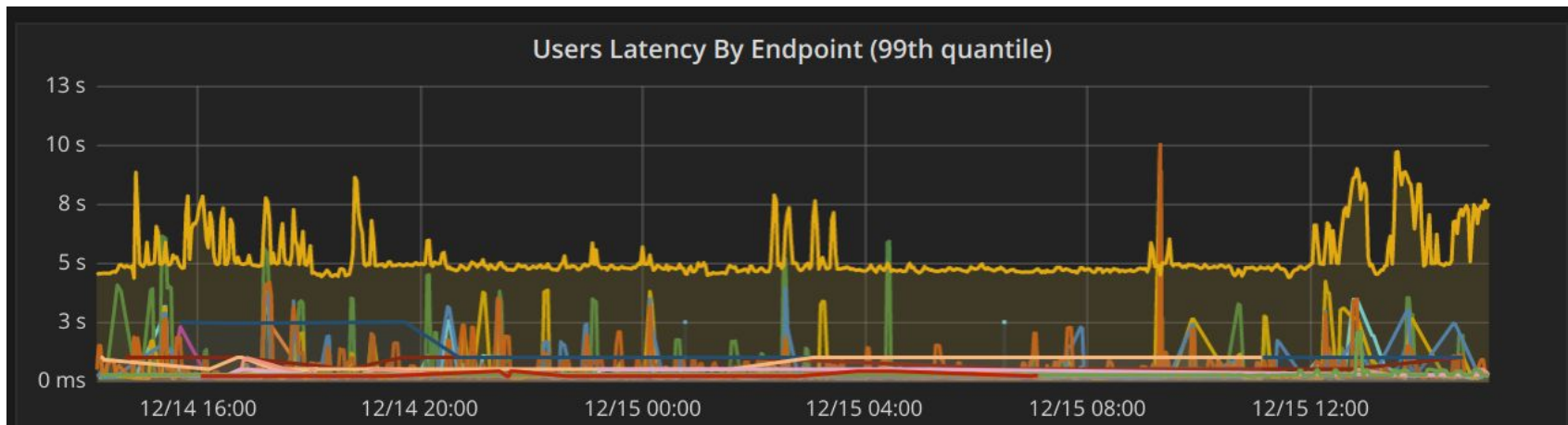
Prometheus Histogram



```
histogram_quantile(0.99,  
  sum(rate(request_duration_bucket[5m])) by (le))
```

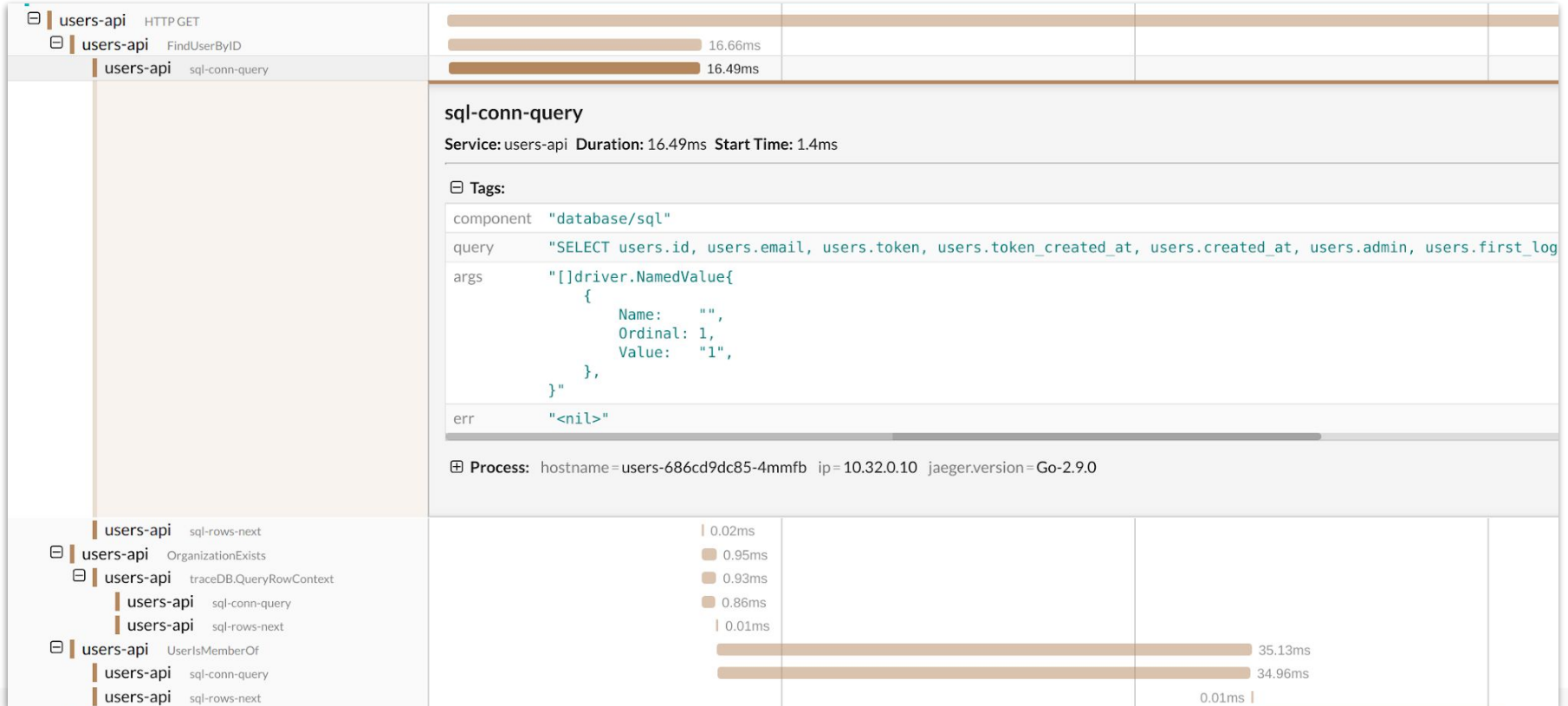


Drilling in

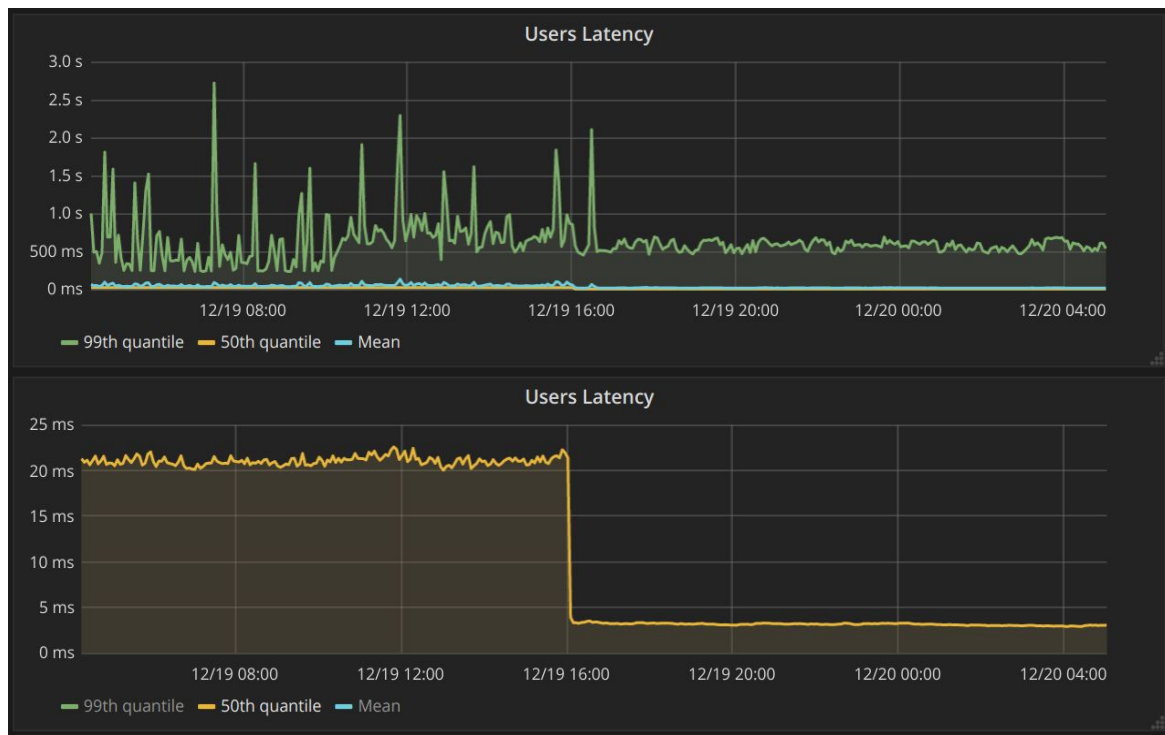


```
histogram_quantile(0.99,  
  sum(rate(request_duration_bucket[5m])) by (le, route))
```


SQL Traces



Fixed it!



Setting up to use Jaeger



Adding Jaeger tracing to a Go program

```
import jaeger "github.com/uber/jaeger-client-go/config"

agentHost := os.Getenv("JAEGER_AGENT_HOST")
cfg := jaeger.Configuration{
    Reporter: &jaeger.ReporterConfig{
        LocalAgentHostPort: agentHost+":6831",
    },
}
closer, err := cfg.InitGlobalTracer(serviceName)
```



Adding the agent host to Kubernetes yaml

```
spec:  
  containers:  
  - name: foo  
    env:  
    - name: JAEGER_AGENT_HOST  
      valueFrom:  
        fieldRef:  
          fieldPath: status.hostIP
```



Adding SQL tracing to a Go program

```
import "github.com/lib/pq"  
import "github.com/ExpansiveWorlds/instrumentedsql"  
  
sql.Register("postgres-i",  
    instrumentedsql.WrapDriver(&pq.Driver{}),  
    instrumentedsql.WithTracer(opentracing.NewTracer()))
```



The Slow Dashboard Query



Some light!

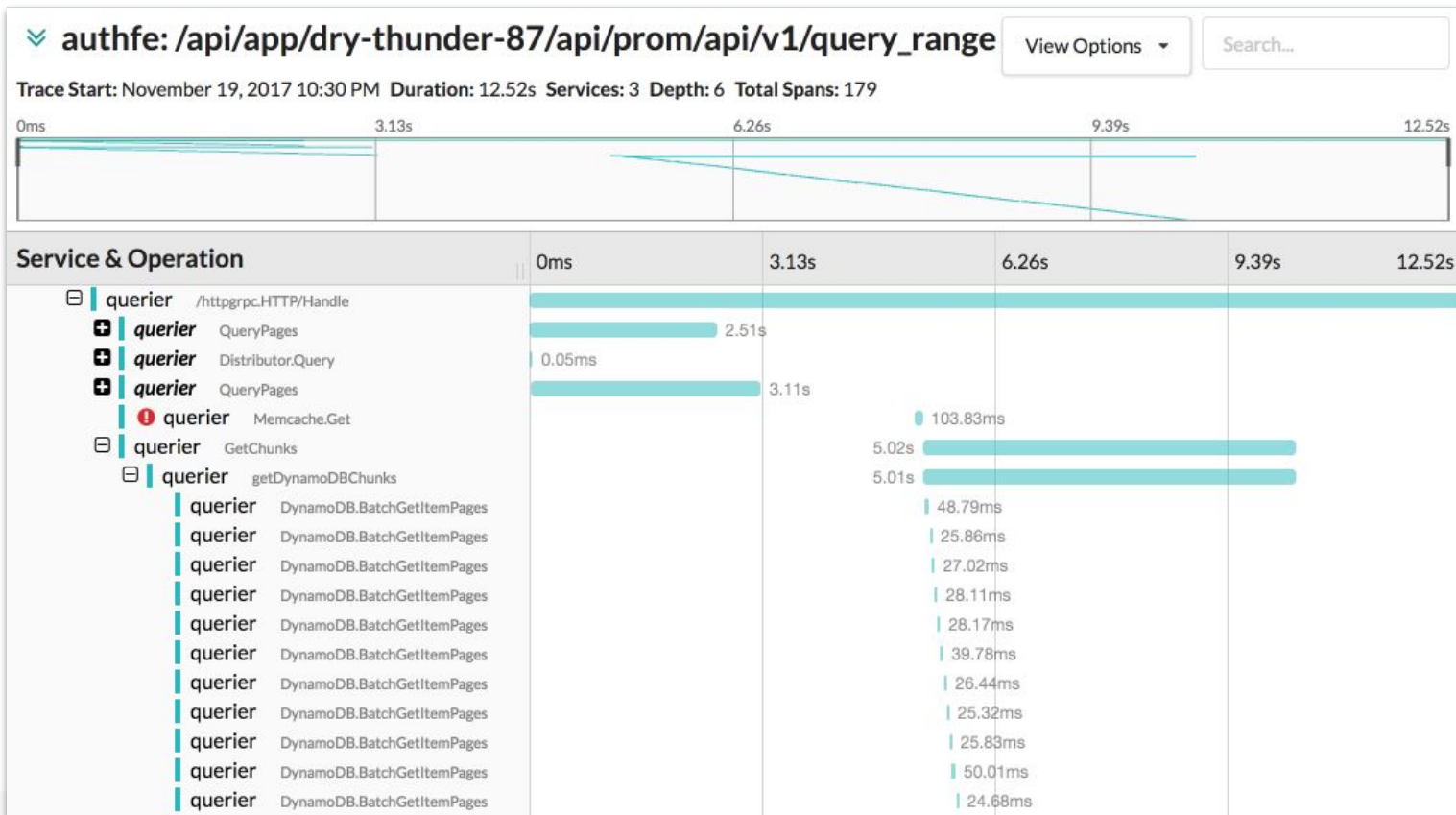


Exhibit A

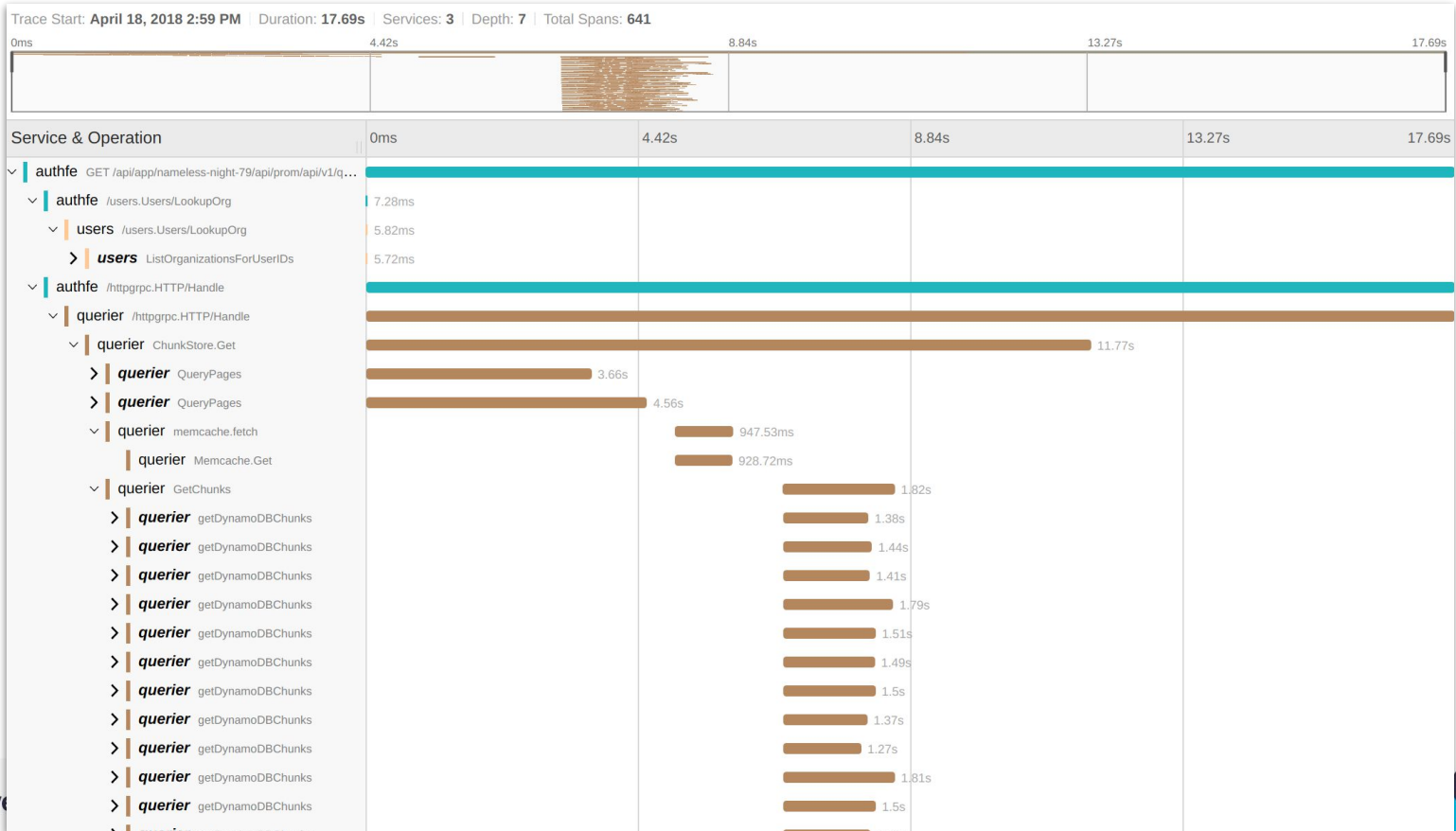
  querier Memcache.Get		 103.83ms
---	--	--



Exhibit B



Parallelised!



Tracing Patterns



Tracing patterns: 0

Look into the span that is longest.



Tracing patterns: 00

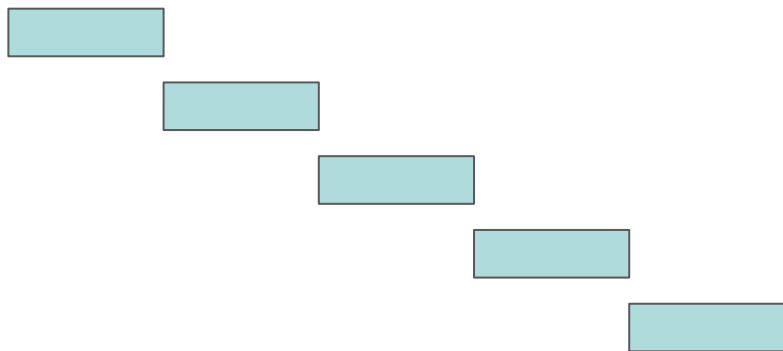
Need another span to tell us what happened in the middle



Tracing patterns: 1

Long diagonal line - each span follows after the one before

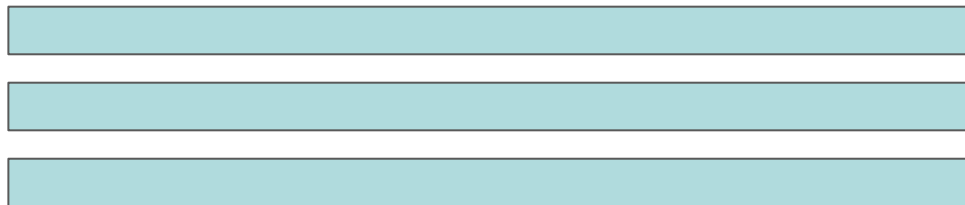
⇒ Look into whether those steps can be parallelised



Tracing patterns: 2

All spans are exactly the same length.

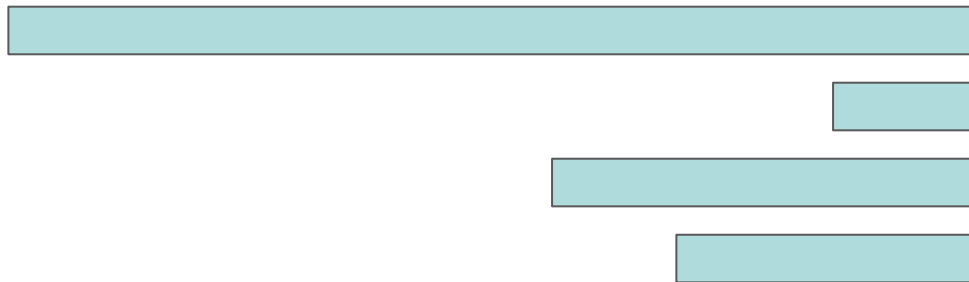
⇒ Look for what is artificially constraining processing, e.g. a timeout.



Tracing patterns: 3

Lots of spans end at exactly the same time.

⇒ Look for an interlock



Takeaways

Measure, Measure, Measure

Prometheus and Jaeger are great tools that repay some work in setting up

Look for those patterns



Questions?

Follow me on Twitter: [@bboreham](https://twitter.com/bboreham)



Weave Cloud: Container Ops-aaS

Combines and extends OSS to offer control, management and automation for teams building containerized applications, as a service.

