Prometheus

What it is, what is new, and what's coming

Julien Pivotto and Richard "RichiH" Hartmann

inuits OPEN SOURCE INNOVATORS

Grafana Labs

Prometheus

What it is

Prometheus 101

- Inspired by Google's Borgmon
- Time series database
- unit64 millisecond timestamp, float64 value
- Over 1000 community-created instrumentation & exporters
- Metrics, not logs
- Dashboarding via Grafana

Main selling points

- Highly dynamic, built-in service discovery
- No hierarchical model, n-dimensional label set
- PromQL: for processing, graphing, alerting, and export
- Simple operation
- Highly efficient

Main selling points

- Prometheus is a pull-based system
- Black-box monitoring: Looking at a service from the outside (Does the server answer to HTTP requests?)
- White-box monitoring: Instrumenting code from the inside (How much time does this subroutine take?)
- Every service should have its own metrics endpoint
- Hard API commitments within major versions

Time series

- Time series are recorded values that change over time
- Individual events are usually merged into counters and/or histograms
- Changing values are recorded as gauges
- Typical examples
  - Access rates to a web server (counter)
  - Temperatures in a data center (gauge)
  - Service latency (histograms)

Super easy to emit, parse & read

```
http_requests_total{env="prod",method="post",code="200"} 1027
http_requests_total{env="prod",method="post",code="400"} 3
http_requests_total{env="prod",method="post",code="500"} 12
http_requests_total{env="prod",method="get",code="200"} 20
http_requests_total{env="test",method="post",code="200"} 372
http_requests_total{env="test",method="post",code="400"} 75
```

Scale

- Kubernetes is ~Borg
- Prometheus is ~Borgmon, but with Monarch APIs
- Google couldn't have run Borg without Borgmon (and Omega and Monarch)
- Kubernetes & Prometheus are designed and written with each other in mind

Scale

- 2,500,000+ samples/second/instance
- 60,000+ samples/second/core
- 16 bytes/sample compressed to 1.36 bytes/sample

The highest we saw in production on a single Prometheus instance were 125,000,000 active times series at once!

Long-term storage

- Two long-term storage solutions have Prometheus-team members working on them
  - Thanos
    - Historically easier to run, but slower
    - Scales storage horizontally
  - Cortex
    - Easy to run these days
    - Scales storage, ingester, and querier horizontally
- Both converge on tech again; I have annoyed people with “Corthanos” for years

Prometheus

What is new?

Service discoveries

- In the last year, we added 5 new service discoveries
  - DigitalOcean
  - Scaleway
  - Hetzner
  - Eureka
  - Docker
  - Docker Swarm

Basic Authentication / TLS

- Prometheus has gained support for TLS/basic auth server side
- A new “exporter toolkit” has been created for the Go exporters
- TLS/Basic auth is being added to more and more exporters

PromQL

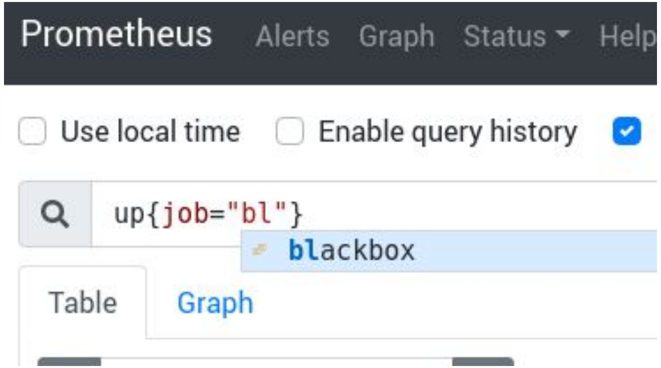
- New functions, like last_over_time
- @ modifier
  - rate(container_cpu_usage) and
    topk(4, rate(container_cpu_usage[5m])) *@ end()*
- Negative offsets
  - up offset -5m
- Composite durations, e.g. 1h30m.

Remote Write receiver

- Prometheus can receive metrics from Remote Write
- Writing from one Prometheus server to another
- Enable new use cases, like Prometheus “on the edge”

UI

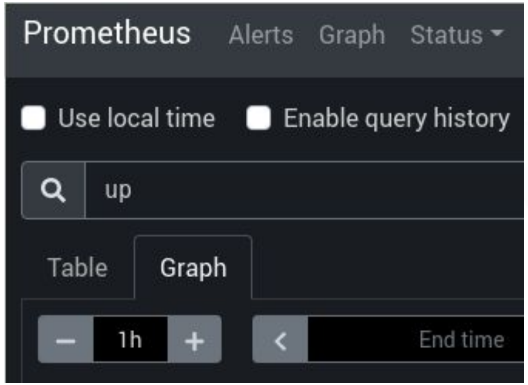
- Prometheus has switched to the React UI by default
- New fully-featured editor, with labels autocompletion, snippets
- Dark theme



□ Use local time □ Enable query history ☑

up{job="bl"}

blackbox



□ Use local time □ Enable query history

up

1h

Exemplars

```
http_request_seconds{le=”5.0”} 9036.32 # {trace_id="KOO5S4vxi0o"} 0.67
```

- Attach external data to metrics (trace ID)
- Easily jump from metrics to traces
- Grafana supports them
- Trace & span ID format taken from W3C tracing specs

Alertmanager

- Time-based muting
  - Do not send alerts in weekends / Out of business hours
  - Controlled per route
- Negative matchers
  - Silence alerts that do not match certain labels

Prometheus

What is coming

Aggressively open

- Historically, Prometheus has been conservative even with features marked EXPERIMENTAL
  - We treated them as stable
- Revisiting a lot of old assumptions, and enabling more use cases
- Make our code more modular, easier to re-use
- Lots of work on Agents and data pipelines; support all deployment and operating models in upstream https://github.com/prometheus
- Mixins out of the box

Aggressively open

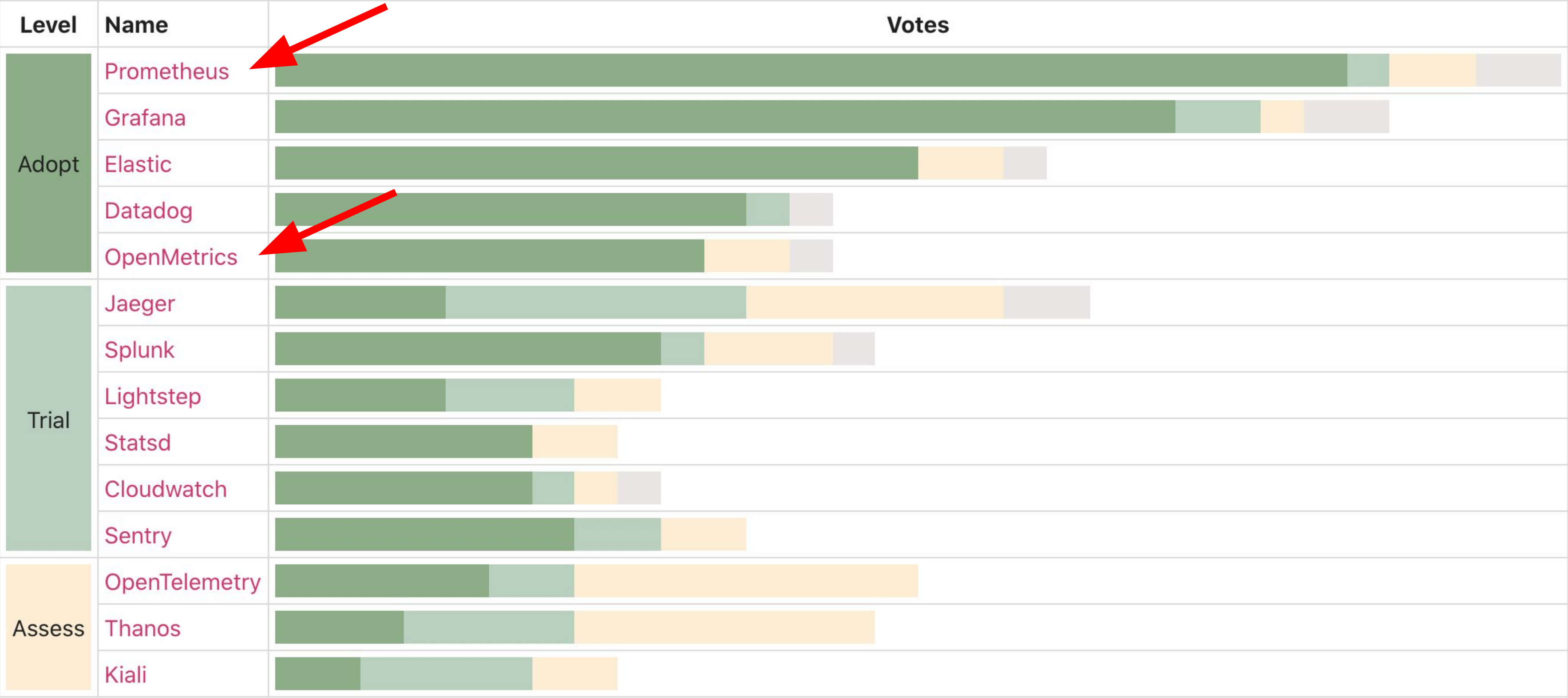
- Current design docs
  - Prometheus Agent
  - Configuration handling in exporters and Prometheus 3.x
  - Improved joins in PromQL across different label names

“

Imitation is the sincerest form of flattery

Oscar Wilde

CNCF End User Survey on Observability



Level	Name	Votes
Adopt	Prometheus	
	Grafana	
	Elastic	
	Datadog	
	OpenMetrics	
Trial	Jaeger	
	Splunk	
	Lightstep	
	Statsd	
	Cloudwatch	
	Sentry	
Assess	OpenTelemetry	
	Thanos	
	Kiali	

sysdig

"support the Prometheus query language, PromQL"

Jun 2020

aws

"AWS Introduces … Amazon Managed Service for Prometheus"

Jan 2021

OpenTelemetry

Working towards full compatibility

Mar 2021

WAVEFRONT by vmware

"added support for PromQL"

Sept 2020

New Relic.

"PromQL-style syntax"

Jul 2020

influxdata

"Creating the PromQL Transpiler for Flux"

Jun 2019

Timescale

"Promscale supports … PromQL"

Oct 2020

Tests, compliance, and compatibility

- CNCF, vendors, and projects have asked us for support
  - OpenMetrics specification
  - Prometheus Remote Write Specification v0.1
- Test suites
  - PromQL Compliance
  - Remote Write Compliance
  - OpenMetrics Compliance
  - Potentially TSDB & data correctness

Share your thoughts!

- All dev summits recorded and open to join
  - Prometheus dev summit, rolling document
  - Prometheus Community Meeting
  - Prometheus Storage Working Group
  - Prometheus Contributor Office Hours
- All public meetings in one calendar
- Prometheus Monitoring YouTube Channel

Thank you!

https://twitter.com/roidelapluie
https://twitter.com/TwitchiH