

The Future of Prometheus and its Ecosystem

Richard “RichiH” Hartmann



Recap

What happened since PromCon 2019?

2.14: React UI

Prometheus Alerts Graph Status ▾ Help Classic UI








☐ Enable query history ☐ Use local time

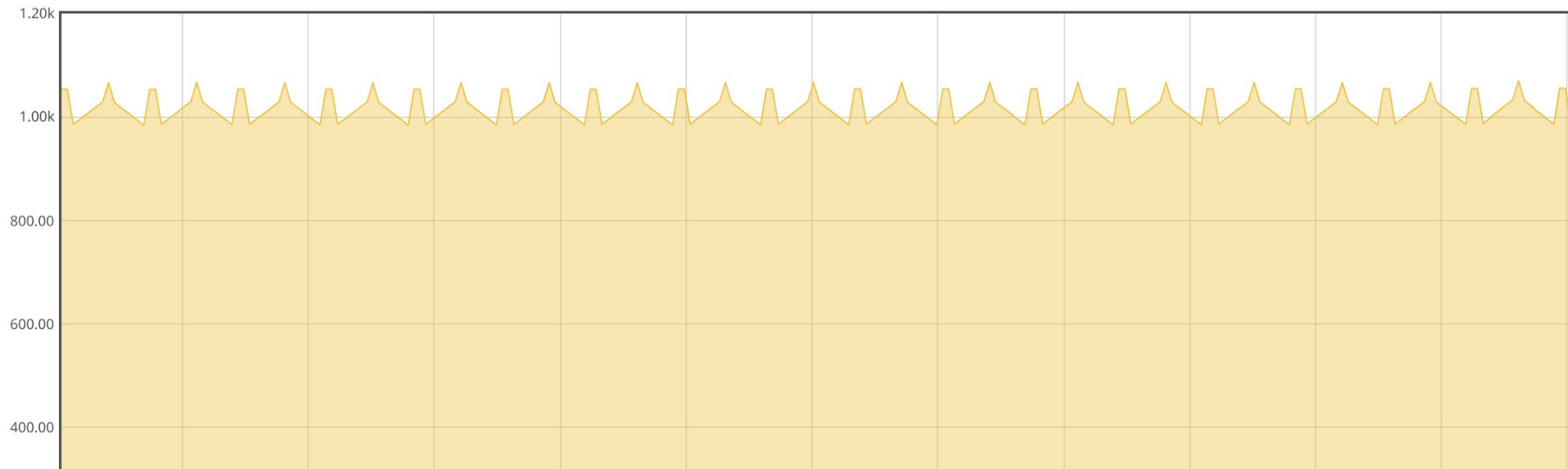
 `sum(rate(process_cpu_seconds_total[2m]))`

Execute

Table Graph

Load time: 149ms Resolution: 14s Result series: 1

 1h   End time   Res. (s)  



2.15: Metadata API

The screenshot shows the Grafana Explore interface with the Prometheus data source selected. The search bar contains the text 'container_fs_read'. A dropdown menu is open, showing a list of metrics. The first metric, 'container_fs_read_seconds_total', is highlighted. To the right of the dropdown, a tooltip for this metric is displayed, showing its type as 'COUNTER' and its description: 'Cumulative count of seconds spent reading'. The interface also includes a 'Split' button, a 'History' button, and a 'Graph' view selector.

Explore prometheus Split History

Metrics + Add query

container_fs_read

History

container_fs_read_seconds_tot...

Metrics

container_fs_read_seconds_tot...

container_fs_reads_bytes_total

container_fs_reads_merged_tot...

container_fs_reads_total

container_fs_read_seconds_total

COUNTER: Cumulative count of seconds spent reading

series. [Show all 1385](#)

Graph

1.0

0.5

2.16: Improvements & stability

[FEATURE] React UI: Support local timezone on /graph #6692

[FEATURE] PromQL: add absent_over_time query function #6490

[FEATURE] Adding optional logging of queries to their own file #6520

[ENHANCEMENT] React UI: Add support for rules page and "Xs ago" duration displays #6503

[ENHANCEMENT] React UI: alerts page, replace filtering togglers tabs with checkboxes #6543

[ENHANCEMENT] TSDB: Export metric for WAL write errors #6647

[ENHANCEMENT] TSDB: Improve query performance for queries that only touch the most recent 2h of data. #6651

[ENHANCEMENT] PromQL: Refactoring in parser errors to improve error messages #6634

[ENHANCEMENT] PromQL: Support trailing commas in grouping opts #6480

[ENHANCEMENT] Scrape: Reduce memory usage on reloads by reusing scrape cache #6670

[ENHANCEMENT] Scrape: Add metrics to track bytes and entries in the metadata cache #6675

[ENHANCEMENT] promtool: Add support for line-column numbers for invalid rules output #6533

[ENHANCEMENT] Avoid restarting rule groups when it is unnecessary #6450

2.17: Isolation

Port isolation from old TSDB PR #6841

 **Merged** beorn7 merged 2 commits into `master` from `beorn7/isolation`  on Feb 28

 Conversation 142

 Commits 2

 Checks 1

 Files changed 4



beorn7 commented on Feb 18

Member



The original PR was [prometheus-junkyard/tsdb#306](#) .

I tried to carefully adjust to the new world order, but please give this a very careful review, especially around iterator reuse (marked with a TODO).

On the bright side, I definitely found and fixed a bug in txRing.

2.18: Tracing & exemplars

Cortex / Reads ▾

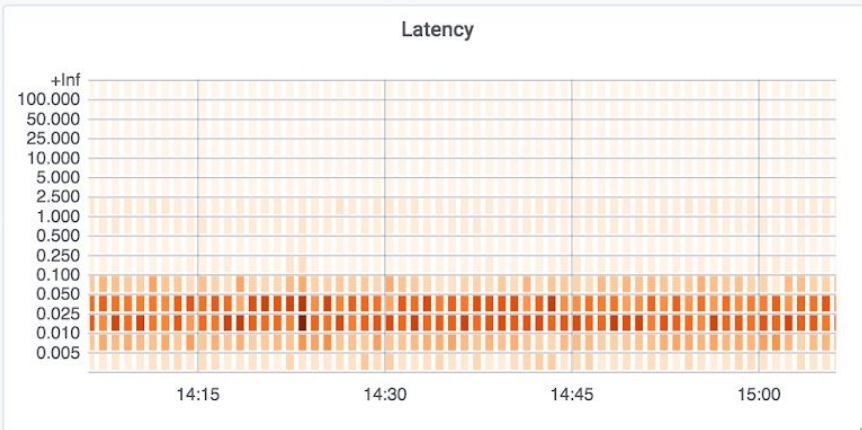
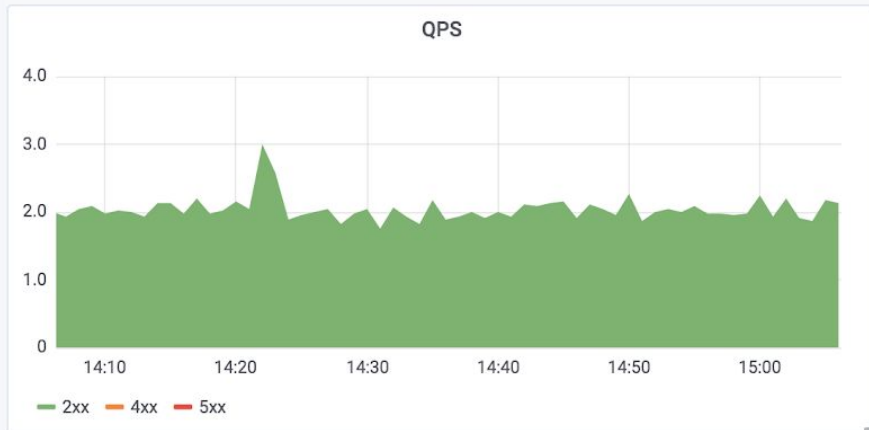


datasource ops-tools1 ▾

cluster All ▾

namespace All ▾

Frontend (cortex_gw) ▾

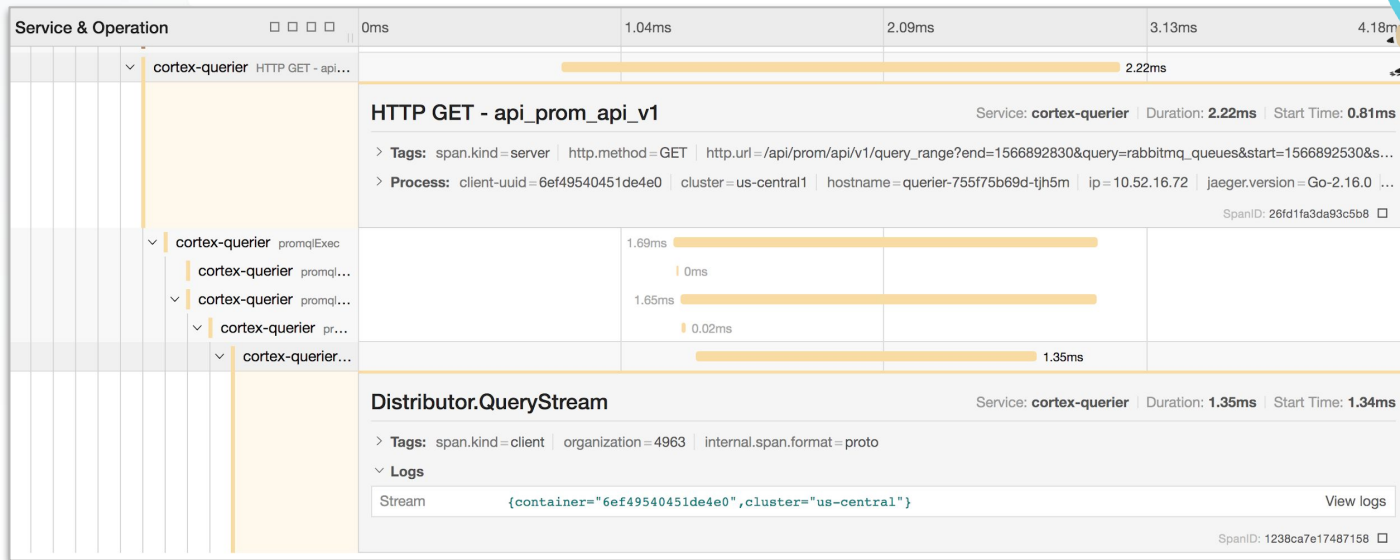


Query Frontend ▾

2.18: Tracing



JAEGER



2.19: RAM!



2.20

[CHANGE] promtool: Changed rule numbering from 0-based to 1-based when reporting rule errors. #7495

[CHANGE] Remote read: Added prometheus_remote_storage_read_queries_total counter and prometheus_remote_storage_read_request_duration_seconds histogram, removed prometheus_remote_storage_remote_read_queries_total counter.

[CHANGE] Remote write: Added buckets for longer durations to prometheus_remote_storage_sent_batch_duration_seconds histogram.

[CHANGE] TSDB: WAL compression is enabled by default. #7410

[FEATURE] PromQL: Added group() aggregator. #7480

[FEATURE] SD: Added Docker Swarm SD. #7420

[FEATURE] SD: Added DigitalOcean SD. #7407

[FEATURE] SD: Added Openstack config option to query alternative endpoints. #7494

[ENHANCEMENT] Configuration: Exit early on invalid config file and signal it with exit code 2. #7399

[ENHANCEMENT] PromQL: without is now a valid metric identifier. #7533

[ENHANCEMENT] PromQL: Optimized regex label matching for literals within the pattern or as prefix.

[ENHANCEMENT] promtool: Added time range parameters for labels API in promtool. #7469

[ENHANCEMENT] Remote write: Include samples waiting in channel in pending samples.

[ENHANCEMENT] Scrape: Ingest synthetic scrape report metrics atomically.

[ENHANCEMENT] SD: Reduce timeouts for Openstack SD. #7507

[ENHANCEMENT] SD: Use 10m timeout for Consul watches. #7508

[ENHANCEMENT] SD: Added AMI meta label for EC2 SD.

[ENHANCEMENT] TSDB: Increment WAL corruption counter.

[ENHANCEMENT] TSDB: Improved query performance.

[ENHANCEMENT] UI: Display data source.

[ENHANCEMENT] UI: Improved error messages.

[ENHANCEMENT] UI: Improved error messages.

[BUGFIX] API

[BUGFIX] API: Fixed error when querying metrics. #7346

[BUGFIX] API: Fixed error when querying metrics. #7346

[BUGFIX] API: Fixed error when querying metrics. #7346

[BUGFIX] API: Fixed error when querying metrics. #7346

[BUGFIX] API: Fixed error when querying metrics. #7346

[BUGFIX] API: Fixed error when querying metrics. #7346

[BUGFIX] API: Fixed error when querying metrics. #7346

[BUGFIX] API: Fixed error when querying metrics. #7346

[BUGFIX] API: Fixed error when querying metrics. #7346

[BUGFIX] API: Fixed error when querying metrics. #7346

node_exporter 1.0 & TLS

- node_exporter reached 1.0
- Has EXPERIMENTAL TLS support
- CNCF is sponsoring a review by Cure53
 - Initial results look good
 - We will publish the report in full

Future

Prometheus is not feature complete

- I sometimes get the feeling that we, as a project, are resting on our laurels
- Some time ago, I started a Grafana-internal brainstorming with the motto “Prometheus is not feature complete” and asked Red Hat to do the same
- Widely-scoped brainstorming document about ALL the things
- Basis for discussion, specific topics broken into dev summit discussion points

Prometheus dev summit

- COVID etc. made us not have one up to now
 - Solution: Shorter, more often, and virtual
 - 4 hour blocks of dev summit instead of 1-2 full days at once
- First virtual one was on 2020-07-10
- Next one will be on 2020-08-07
- Will keep doing them until backlog is empty
- Lots of backlog, and we keep adding MOAR



Metadata

- Metadata is starting to be useful in Prometheus (see 2.15)
- We need to do more, e.g. propagate through remote read-write
- Lots of interesting questions if metadata changes/flaps

• CONSENSUS: We want to support more metadata. Work will be done in [LINK TO DESIGN DOC](#)

• CONSENSUS: <https://github.com/prometheus/prometheus/pull/6815> is acceptable as an EXPERIMENTAL stopgap. It is likely to change for 3.x

Workflow changes; and s/master/main/

- Some more janitorial work around our workflows
- We're also serious about spearheading verbiage cleanup
 - Not the most important thing we can do, but something we can do
 - Waiting for GitHub to provide tooling
 - Reach out if you want to help clean up, potentially through Community Bridge

- CONSENSUS: Set "Require status checks to pass before merging" (see figure above) on all prometheus/... repos.
- Prevent direct pushes to main branch? Prevent force pushes to main branch?
 - Brian: Java release tools need to be able to push to main branch
 - CONSENSUS: Disable force-pushes to all main branches
 - CONSENSUS: Default behavior is to allow pushes to the main branch, but it should be disabled for certain "important" repos, e.g. prometheus/prometheus, at the discretion of the maintainer.

Backfilling!

- We will start working on backfilling data in Prometheus proper!
- Contentious topic, and a good showcase of how you can reach consensus

• CONSENSUS: We want to support Prometheus text exposition format and OpenMetrics and one well-defined CSV format for importing data

- CONSENSUS: We want to support backfilling over the network at least via blocks which do not overlap with the head block
- NOT CONSENSUS: We want to support backfilling over the network at least via streams which do not overlap with the head block
- CONSENSUS: We would like to support backfilling over the network via blocks which do overlap with the head block given proper design

Vendoring...

- ``vendor/`` is the mono-repo reinterpretation of dependency handling

• CONSENSUS: Remove it in node_exporter, see if we like it

Janitorial: Mailing lists & IRC

- China blocks Google, and we want to adapt our documentation to make it easier for Chinese users to participate
- prometheus-users+subscribe@googlegroups.com works

- CONSENSUS
 - See if it is possible to subscribe to our MLs without a Google account & enable this; still require subscription for posting
 - Link to public ML archive in our docs/community section, and explain how to subscribe without Google account
 - Disable identification requirements in IRC channels to make using Matrix easier; revert if there are any attacks

Documentation

“The first thing I disliked about Prometheus in 2015 was the documentation, in 2020 I despise it. It’s not human-friendly and almost useless if you don’t already know what you’re doing or at least have all the concepts down.”

- Me, during the dev summit

- CONSENSUS:
 - We will restructure the documentation into three sections
 - User manual, default, low threshold for contribution
 - Reference, what we currently have with PromQL examples of in/output
 - Guides, preferably on a modular basis
 - We will ask Diana Payton for a style guide, overall concept, etc. for all documentation
 - We will try not to break links unless it really makes sense

Read-it-yourself

- The dev summit notes are at https://docs.google.com/document/d/1yuaPKLDvhJNXMF1ubs00m5kE2_6dvCxHowBQIDs0KdU/edit# and being merged to <https://prometheus.io/community/> as we speak

Ecosystem

Exporters, exporters, exporters

- Thousands of exporters worldwide
- We want you to focus on metrics, not on writing HTTP servers
- <https://github.com/prometheus/exporter-toolkit>



Siblings

- Cortex & Thanos
 - Would not exist without Prometheus
 - They cover use cases out of scope for Prometheus
 - Huge maintainer overlap
- Sandbox -> Incubation
 - Both passed CNCF SIG Observability review & due diligence
 - Both passed CNCF TOC due diligence
 - Public comment period on TOC mailing list
 - Formal vote by TOC soon

CNCF SIG Observability

- There's a new SIG in town
- Now that no direct action items from TOC are open, we're free!
 - Best Current Practice documents
 - Probably focus on metrics first
 - Data analysis
 - Currently defining use cases
- Join us!
 - Call is every second Tuesday, 16:00 UTC
 - cncf-sig-observability@lists.cncf.io
 - #sig-observability on <https://slack.cncf.io/>

OpenMetrics

- Making good progress on tightening up prose into IETF ID
- Exemplars in Prometheus (and Cortex & Thanos) are a direct result of OpenMetrics
- I joined the OpenTelemetry calls some months ago to ensure interop
 - OM is on track to be a first-class exposition format of OTel
 - Prometheus experience and OM discussions proved useful to OTel

Loki

- When I started OpenMetrics in 2016, I stated I wanted to tackle
 - Metrics
 - Logs
 - Traces, in this order
- Many discussions in the background over the years
- Very excited to see Loki take shape
 - Same data model for metrics & logs
 - Seamless transition between the two
 - Loki will support extraction of metrics from logs in the next version

Ecosystem, too



In tech, we tend to take the view that politics should be “kept out”.

This is wrong. Open Source is fundamentally a political movement, a movement to literally change society, to create equality and freedom.



Not seeing the need to address all these concerns is a position coming from a fundamental misunderstanding of society; and of privilege.



Resist racism

Resist fascism

Black lives matter

Thanks,
Q&A,
follow me on Twitter @TwitchiH,
my SoundCloud is “Prometheus”