Quo vadis, Prometheus?

Monitoring at scale

Richard Hartmann,
RichiH@{freenode,OFTC,IRCnet},
richih@{fosdem,debian,richih}.org,
richard.hartmann@space.net

2017-11-16

‘whoami‘

- Richard ”RichiH” Hartmann
- Swiss army chainsaw at SpaceNet
  - Currently responsible for building one of the most modern datacenters in Europe
  - ...and always looking for nice co-workers in the Munich area
- FOSDEM, DebConf, DENOGx, PromCon staff
- Author of `https://github.com/RichiH/vcsh`
- Debian Developer
- Prometheus team member

Show of hands

- Who has heard of Prometheus?
- Who is considering to use Prometheus?
- Who is POCing Prometheus?
- Who uses Prometheus in production?

Prometheus 101

- Inspired by Google's Borgmon
- Time series database
- int64 timestamp, float64 value
- Instrumentation & exporters
- Not for events
  - Logging
  - Tracing
  - etc.
- 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

Cloudy with a chance of buzzwords

- So it's built with highly dynamic environments in mind
- Containers, sidecars, microservices, ALL the cloud
- But it's a monolithic application

- ...why?

Resilience, resilience, and also resilience

- What do you need for operations?
  - Power and cooling
  - Network connectivity
  - Monitoring
- The rest you can fix

All new and shiny 2.0

- Released on 2017-11-08
  - (Yes, I did suggest we release today along with this talk)
- Various cleanups and improvements
- See
  `https://github.com/prometheus/prometheus/releases`

Three main features

- Storage backend
  - Caveat: Prometheus 2.0 comes with storage v3
- Staleness handling
- Remote read & write API is now stable-ish
- Links to in-depth talks about these features are at the end

Prometheus 1.x

- We used to have one file per time series
- Relatively easy to implement
- Pretty efficient
- Why change?

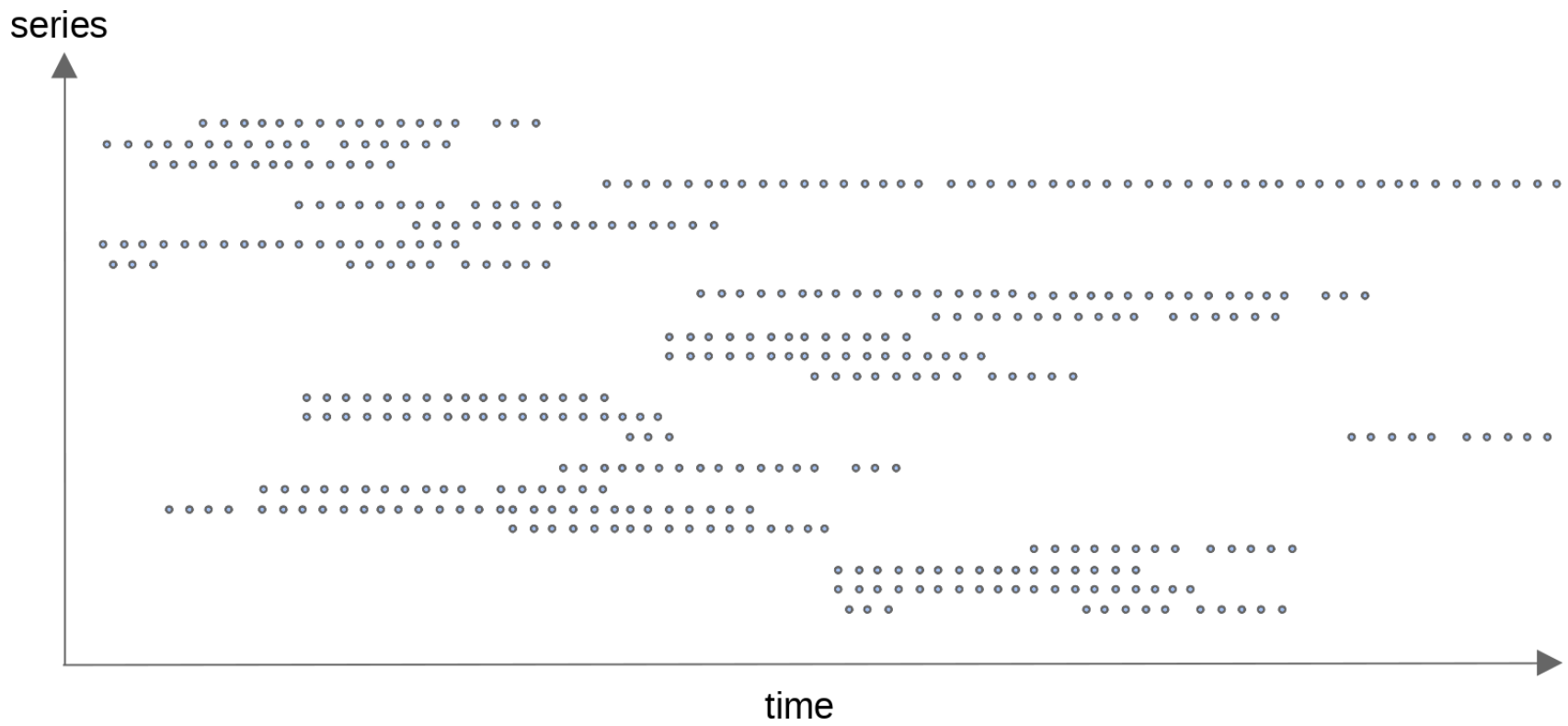
Churn

- Churn was becoming more and more of a problem
- There's a company with a 15 minute maximum lifetime for their containers
- If you have a lot of files which might contain data for any given time frame, you need to look at all of them

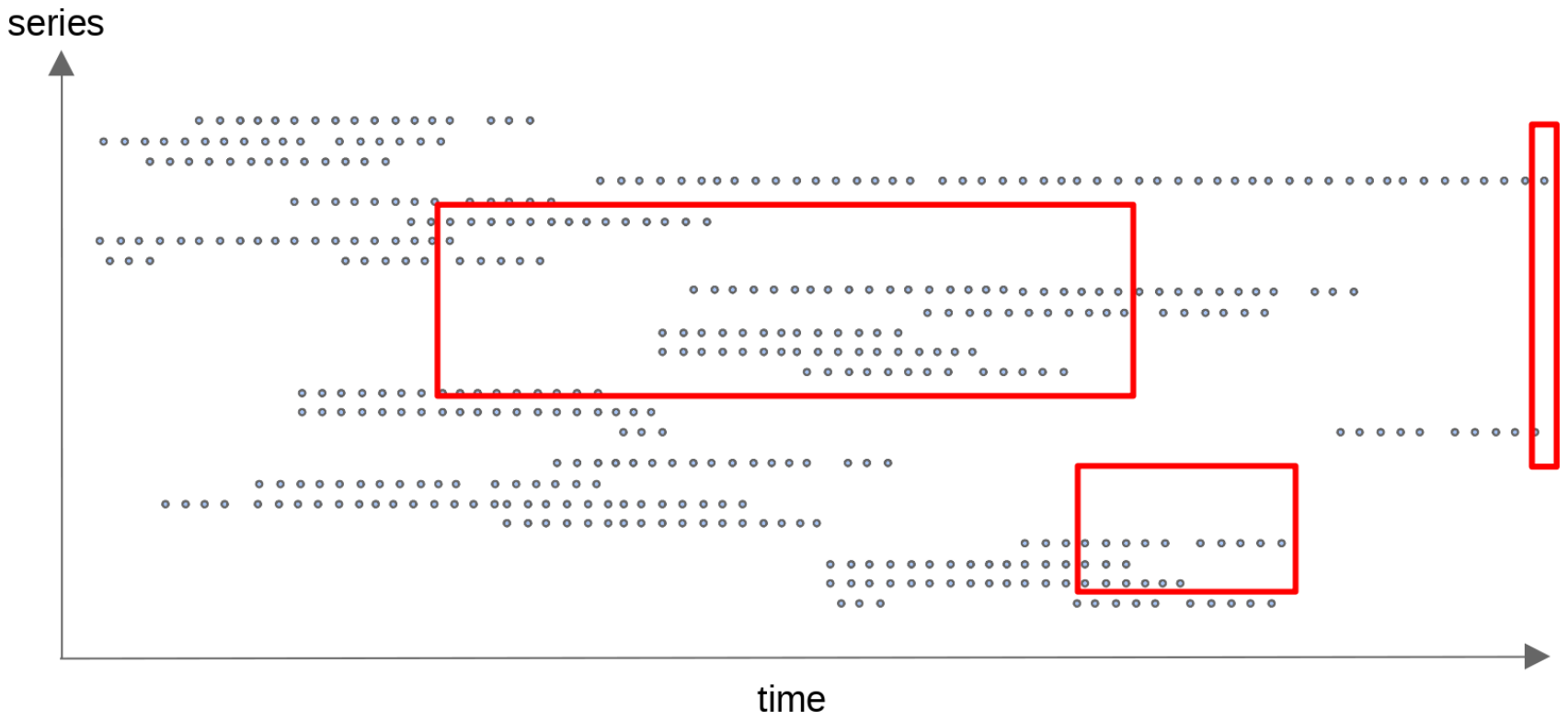
Storage v3

- Fabian Reinartz had an idea about new storage
- This POC turned out to be so good, we decided to cut a major release for it
- How does it work?

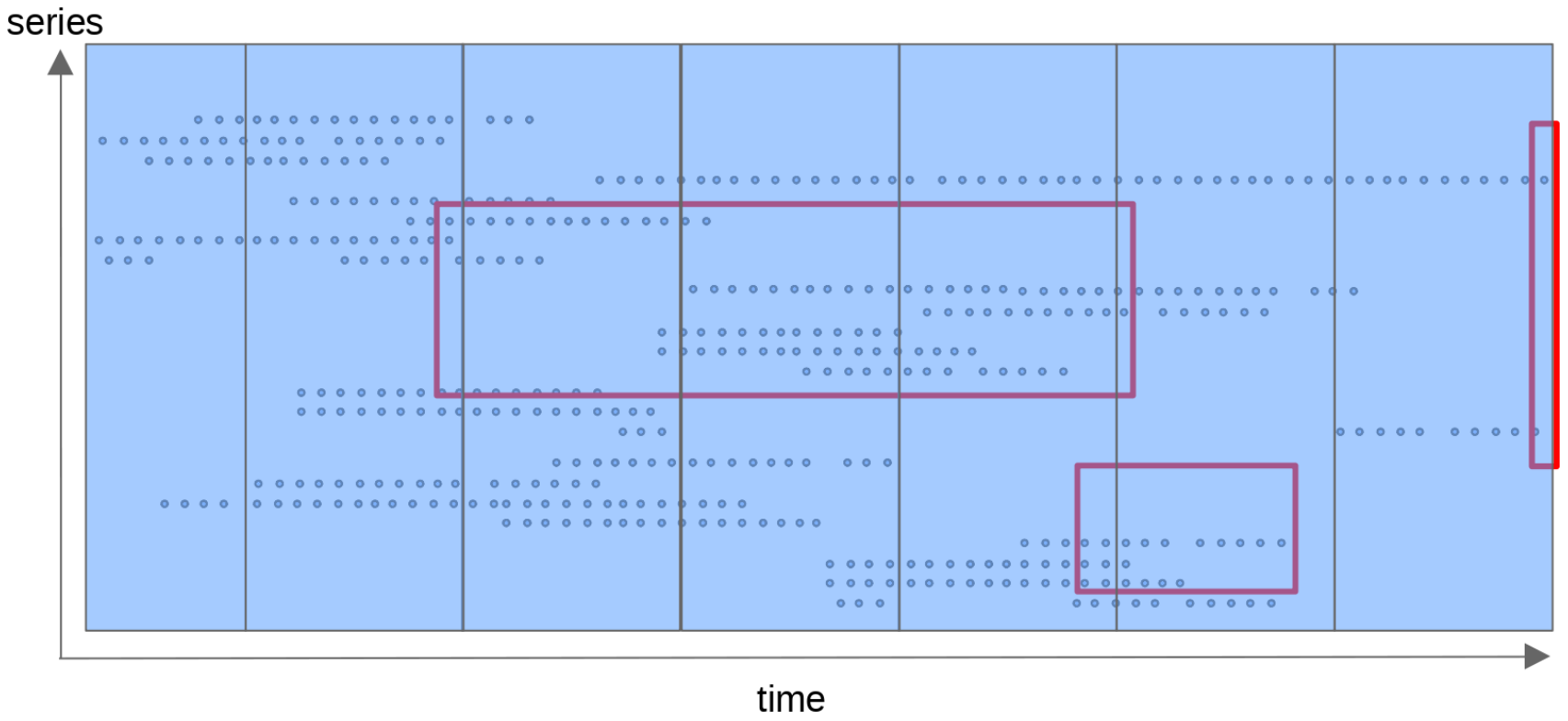
One file per series



Selection



Blocks



Storage v3

- Deletions set tombstones
- Actual deletion done via compaction runs, or triggered by large tombstone ratio
- Oh, and we now simply mmap storage blocks into RAM and let the OS handle the rest

Test setup

- Kubernetes cluster with dedicated Prometheus nodes
- 800 microservice instances and Kubernetes components
- 120k samples/sec
- 300k active time series
- Swap out 50% of all pods every 10 minutes

Results

- 15x reduction in memory usage
- 6x reduction in CPU usage
- 80-100x reduction in disk writes
- 5x reduction in on-disk size
- 4x reduction in query latency on expensive queries
- Want to reproduce?
  `https://github.com/prometheus/prombench`

One more thing...

Also, you can now properly snapshot and backup Prometheus.

Downside of handling churn

- So now we can handle extreme churn
- ...and suddenly, five minutes staleness timeouts seem awfully long
  - Down alerts continue to fire
  - Double counting
  - Other icky corner cases
- ...and what if you need more than five minutes scrape interval?

Results

- When a target goes away, its time series are considered stale
- When a target no longer returns a time series, it is considered stale
- Longer eval/scrape intervals are now easier to handle

The way there

- The actual technical background is too complex for this talk
- Find recording and slides by Brian Brazil at the end of this talk

Playing nicely with others

- We now have a stable-ish remote read/write API
- Which we're already using ourselves; it's the recommended upgrade path from 1.x
  - You need to upgrade to 1.8.2 or later for this to work

So, about backfill and explicit timestamps...

- If explicit timestamps were icky before, this has now become worse
- You can not ingest data older than the age of the current storage block, nor data much newer
- Staleness vs timestamps is non-trivial

ACID databases...

- **A**tomicity - we have that
- **C**onsistency - we have that
- **I**solation - here, there be 2.1 features
- **D**urability - we have that since 2.0

Isolation

- Each append action gets a write ID (64 bit monotonic counter)
- Every sample's write ID is noted along with value and timestamp
- Any append action which has not yet committed or rolled back is ignored at query time
- We keep write IDs in memory; if we restart or crash, the atomicity of the write ahead log will protect us

Object storage & true HA

- So now all our storage is in self-contained blocks
  - ...this sounds a lot like objects
  - ...so you store them in a S3 interface
  - And suddenly, your local disk is merely a hot cache
- Two implementations
  - `https://github.com/gouthamve/agni`
  - `https://github.com/improbable-eng/thanos`
- Yes, this is a nice way to do long term storage
- Yes, you will be able to splice data from different Prometheus instances together

Humble aspirations

- When we say that we want to change how the world does monitoring, we mean it
- One of our most powerful features are labels
- Labels are encoded in our exposition format
- Some third-party projects and vendors have an issue with supporting a ”competing” project
- Big shout-out to Paul Dix and InfluxData for adopting Prometheus concepts!

OpenMetrics

- We are spinning out Prometheus' exposition format
- It will join CNCF as a full project
- We will submit an ID and try to get an IETF RFC published
- ...so you can sneak this random RFC into the requirements of your next tender ;)
- `https://github.com/RichiH/OpenMetrics`

Already on board

- Cloudflare
- GitLab
- Google
- Grafana
- InfluxData
- Kausal.co
- Oath.com / Yahoo / Verizon
- RobustPerception
- SpaceNet
- Uber

PromCon 2018

- Current plan is to do it in Munich or Dublin
- We will continue to run the conference ourselves, but CNCF is handling money for us
- If you want to sponsor a venue, do let us know
- We can only accept a venue if a team member is living nearby and is willing to invest the time

Generally speaking...

- Yes, we want to change the world
- Simple and resilient operation of Prometheus remains a core goal
- More project and software integrations... and we're talking to hardware vendors as well
- Supporting tomorrow's 10x scale today

Suggested talk

- For a more in-depth intro to Prometheus, visit this talk
  - Max Inden
  - End-to-end Monitoring with the Prometheus Operator
  - today @ 16:00

Recorded talks

- Storing 16 Bytes at Scale: `https://promcon.io/2017-munich/talks/staleness-in-prometheus-2-0/`
- Staleness and Isolation in Prometheus 2.0: `https://promcon.io/2017-munich/talks/staleness-in-prometheus-2-0/`
- Social aspects of change: `https://promcon.io/2017-munich/talks/social-aspects-of-change/`

Further reading

- Prometheus 2017 Dev Summit:
  `https://docs.google.com/document/d/1DaHFaoOsaZ3MDt9yuuxLaCQg8WGadO8s44i3cxSARcM/edit`
- My other talks: `https://github.com/RichiH/talks/`
- OpenMetrics: `https://github.com/RichiH/OpenMetrics`

Thanks!

Thanks for listening!

Questions?

Email me if you want a job in Munich.

See slide footer for contact info.