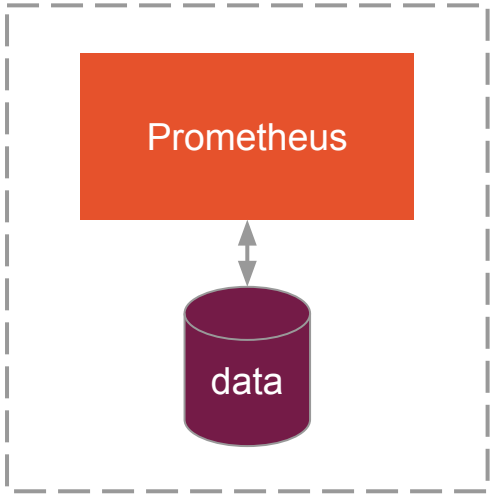
Integrating Long-Term Storage with Prometheus

Julius Volz, March 30, 2017

Local Storage



Good:

- Simple.
- Avoids clustering.
- Reliable with HA setup.

But...

- Not durable.
- Not long-term.
- Not scalable.
- No downsampling etc.
- Not flexible.

Remote Storage

Issue #10

https://github.com/prometheus/prometheus/issues/10

Legacy Write Support

For OpenTSDB, InfluxDB, Graphite

➔ **replace with generic interface!**

Design Questions

Write Path

- Chunks vs. samples?
- Buffering and retries?

Read Path

- Distributed vs. central evaluation?

- protocol and encoding?

Decision

Start simple, go from there.

Write Path

Samples vs. Chunks

Send samples. Prometheus has internal chunking, but...

- What chunk size?
- What about write delay?
- Remote end not always a storage.
- Simplicity of remote end.

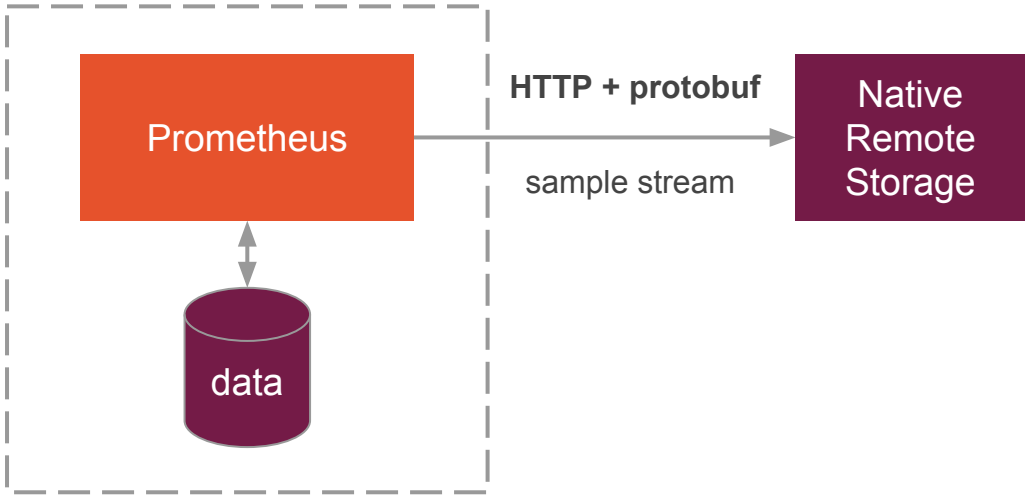
Buffering and retries

- Write shards with dynamic parallelism.
- Minimal queueing and retries (memory!).

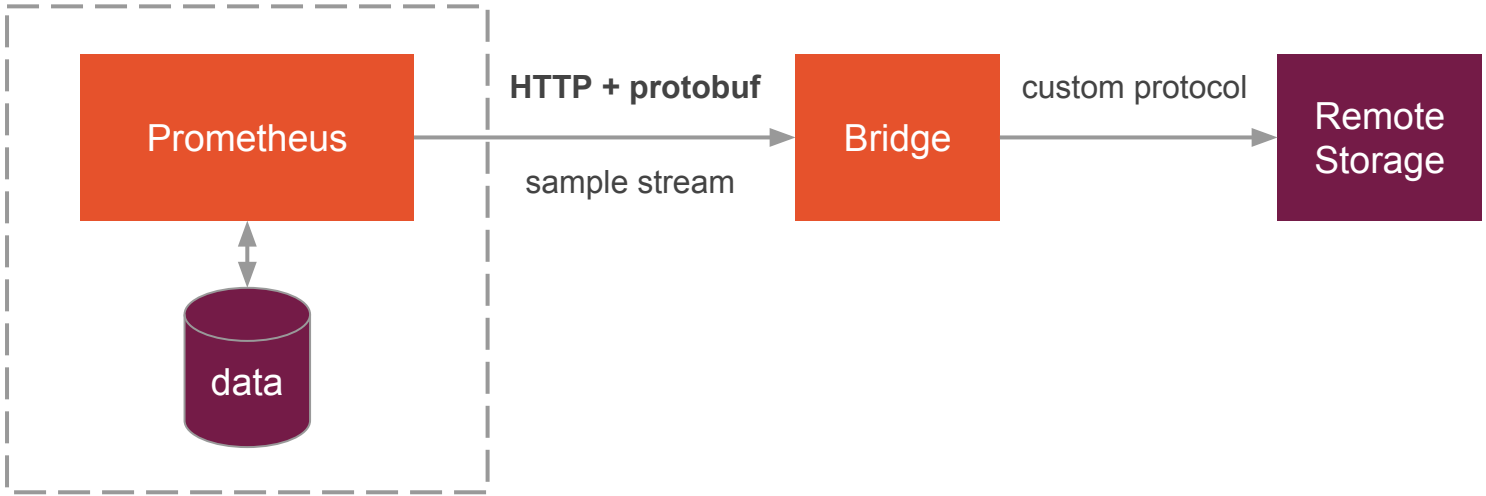
Protocol and Encoding

- HTTP + protobuf.
- Lack of HTTP/2 support “out there”: no gRPC for now.

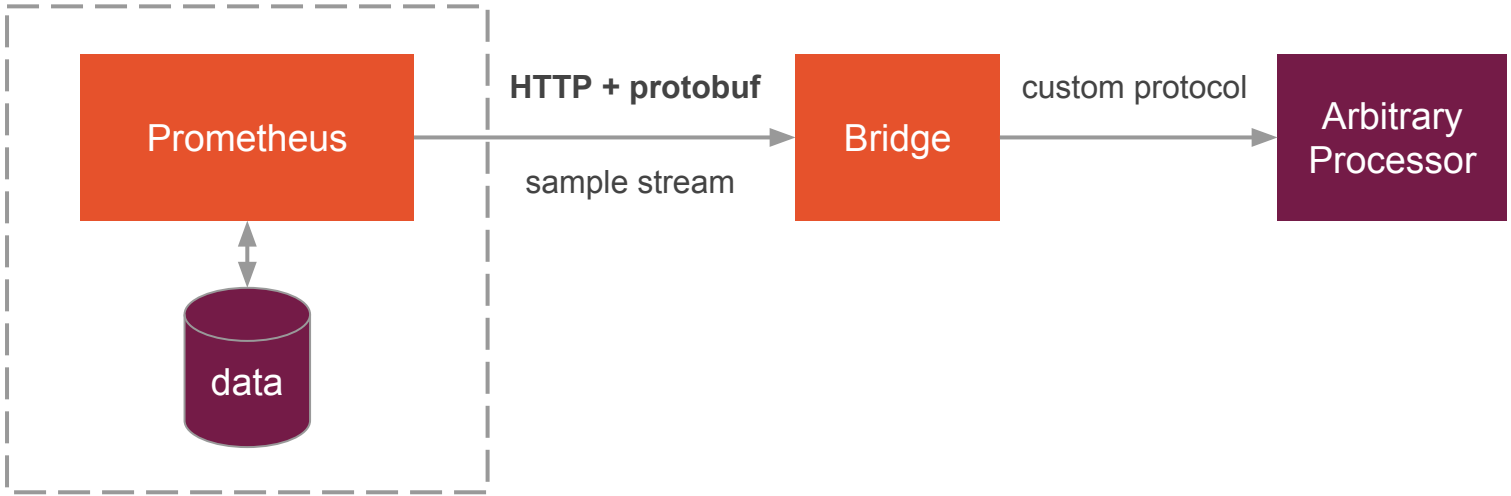
Write Path (Native)



Write Path (Bridge)



Or more generally...



Configuration

```
remote_write:
  - url: "https://my-storage/write"
    write_relabel_configs: <...>
    remote_timeout: <...>
    basic_auth: <...>
    bearer_token: <...>
    proxy_url: <...>
    tls_config: <...>
```

Read Path

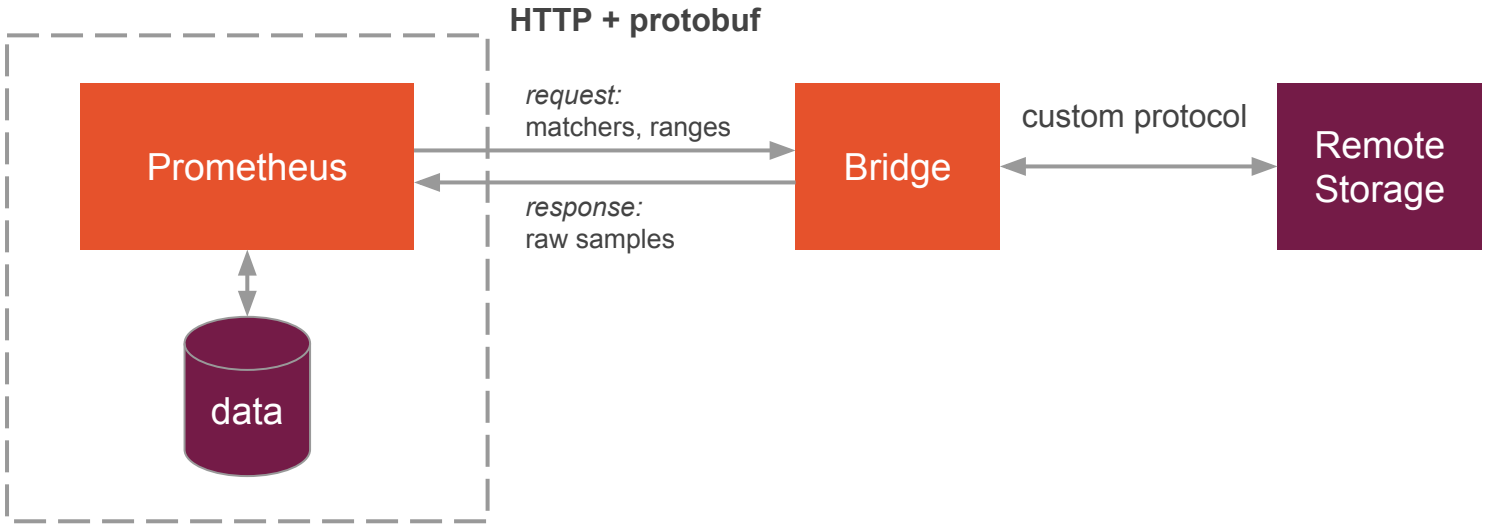
Distributed or Centralized Eval?

Distributed evaluation scales better, but:

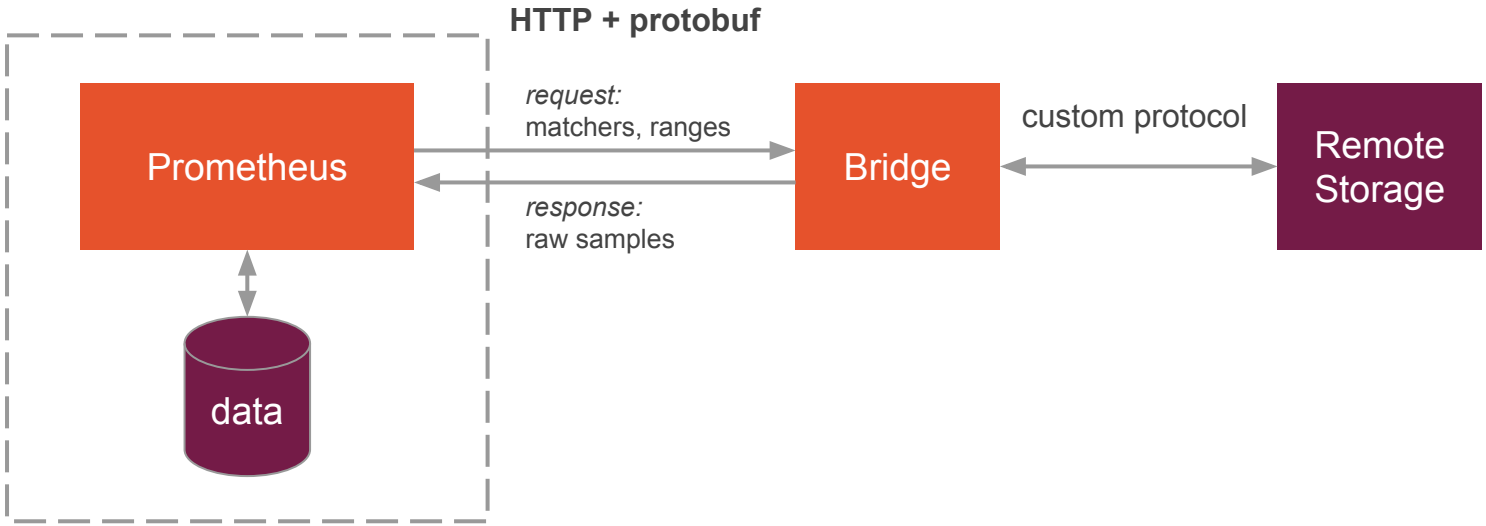
- Requires remote knowledge of PromQL.
- Local eval good enough for most queries.
- Remote downsampling helps.

➔ Centralized PromQL evaluation for now.

Read Path

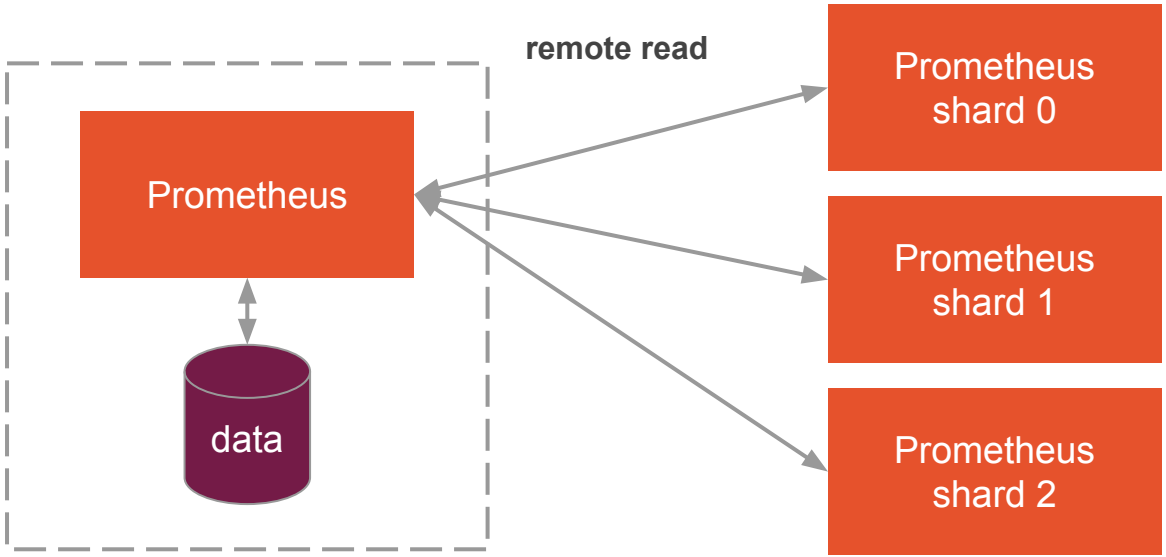

HTTP + protobuf
Prometheus
request:
matchers, ranges
response:
raw samples
Bridge
custom protocol
Remote Storage
data

Read Path



PromQL evaluation in Prometheus!

Future: Query Federation



Configuration

```
remote_read:
  - url: "https://my-storage/read"
    remote_timeout: <...>
    basic_auth: <...>
    bearer_token: <...>
    proxy_url: <...>
    tls_config: <...>
```

Note!

- Rules query only local data.

  Reliability!
- Metadata API only local so far.
- External label magic is applied (TODO).

Real-World Examples

CHRONIX

QAware
Write-only

http://www.chronix.io/
https://github.com/ChronixDB/chronix.ingester

weavecortex

Weaveworks
Read + write

https://github.com/weaveworks/cortex

Real-World Examples

CHRONIX

QAware
Write-only

http://www.chronix.io/
https://github.com/ChronixDB/chronix.ingester

Weaveworks
Read + write

https://github.com/weaveworks/cortex

Weave Cortex

- Horizontally scalable Prometheus.
- As a service or self-hosted (open source).
- Stores data in AWS (GCP support in the works).
- Normal querying API (use with Grafana, etc.).
- Plus generic remote read + write.
- Use and contribute: https://github.com/weaveworks/cortex.

Weave Cortex

DEMO

Build Your Own!

See protobuf definition at:

https://github.com/prometheus/prometheus/blob/master/storage/remote/remote.proto

```
syntax = "proto3";
package remote;

message Sample {
 double value       = 1;
 int64 timestamp_ms = 2;
}

message LabelPair {
 string name  = 1;
 string value = 2;
}

...
```

Beware

This is all still experimental.

Watch for the next release.

Thanks!