Cortex:
Deep Dive

Bryan Boreham (@bboreham)

weaveworks

Tom Wilkie (@tom_wilkie)

Grafana Labs

Show of hands

What is Cortex? (recap)

Demo

Awesome Query Performance

What is Cortex?

Cortex is a time-series store built on Prometheus

- Horizontally scalable
- Highly Available
- Long-term storage
- Multi-tenant

Cortex is a CNCF Sandbox project
https://github.com/cortexproject/cortex

What is Cortex for?

A global view of as many metrics as you need

With **no gaps** in the charts

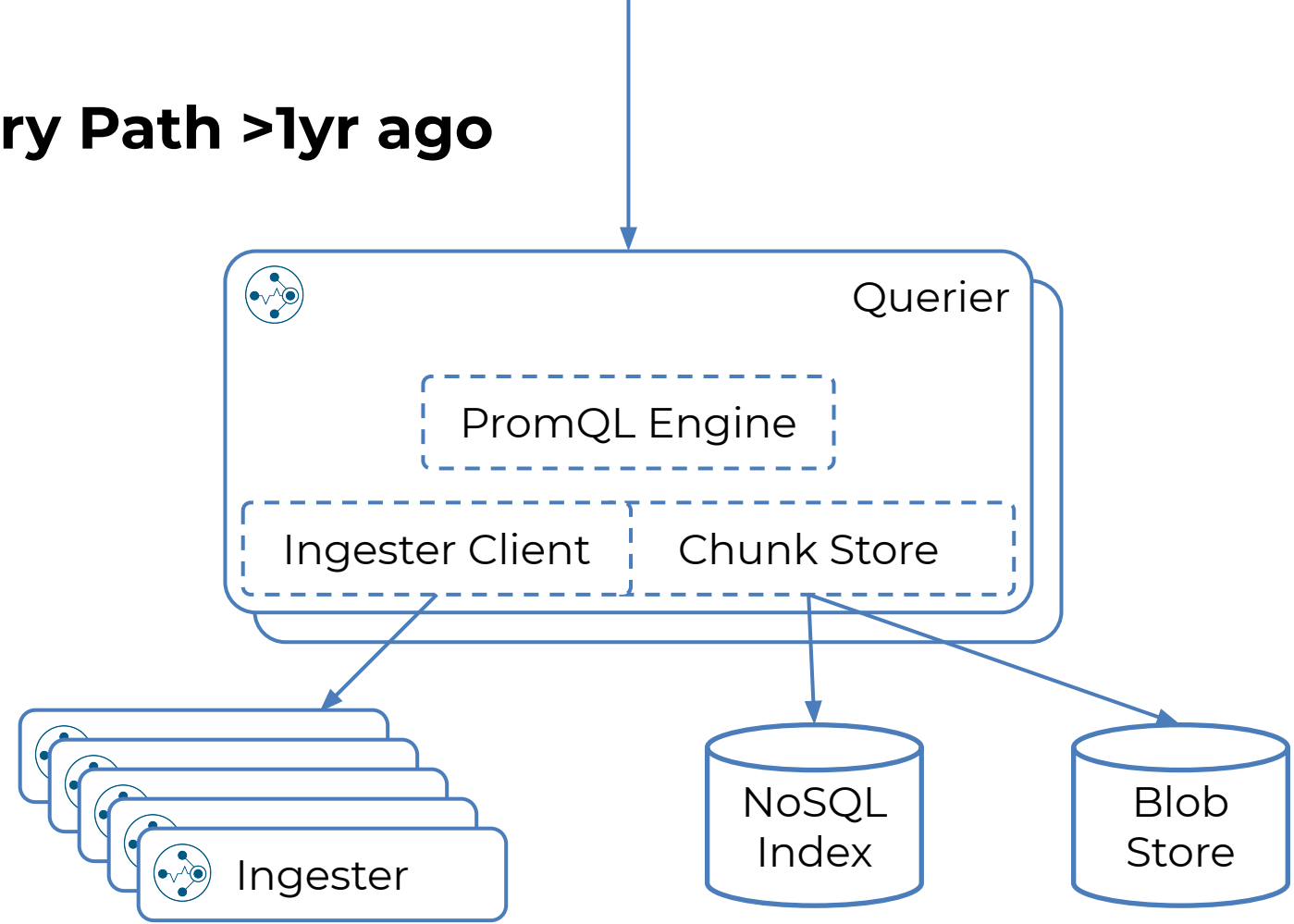
On durable, **long term storage**

Across **multiple tenants**

Demo Time!

Awesome Query Performance

Query Path >1yr ago



Inverted Index

```
rate(http_duration_seconds_count{job="shipping"}[1m])
```

...	
http_duration_seconds_count:job	orders, shipping, customers, ...
http_duration_seconds_count:instance	a, b, c, d, ...
http_duration_seconds_count:path	/foo, /bar, /...
http_duration_seconds_count:result	200, 401, 402, 404, 501, 503, ...
...	

Index Lookup

Suppose PromQL query is:

```
rate(http_duration_seconds_count{job="shipping"}[1m])
```

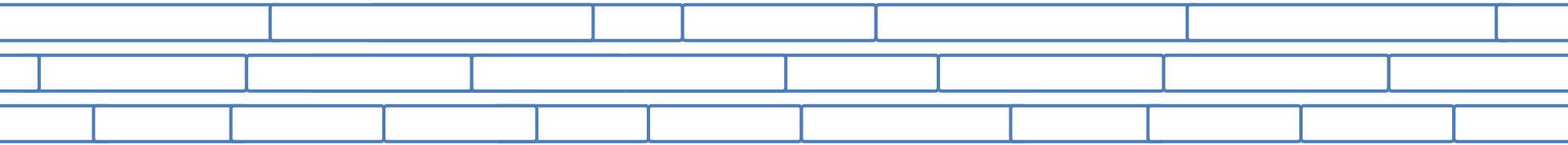
Go to index row `http_duration_seconds:job`

Look up “shipping”

- ➢ set of timeseries
  - ➢ look up each timeseries
    - ➢ set of chunks

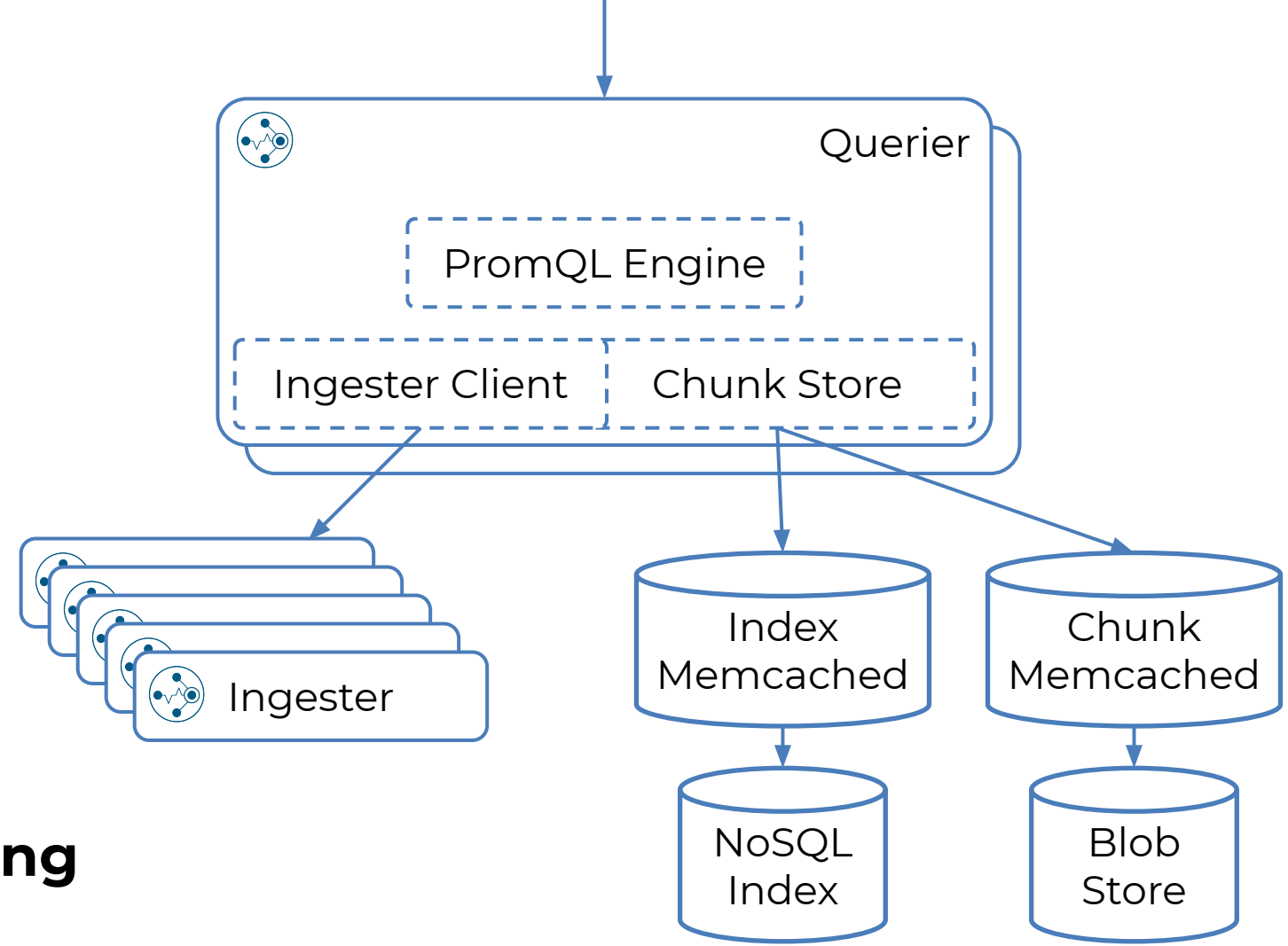
Merging / Deduping samples

- Cortex writes three replicas of each chunk.
- Due to jitter, transfers, outages and replication these won't be identical.
- Have to dedupe and merge at query time,

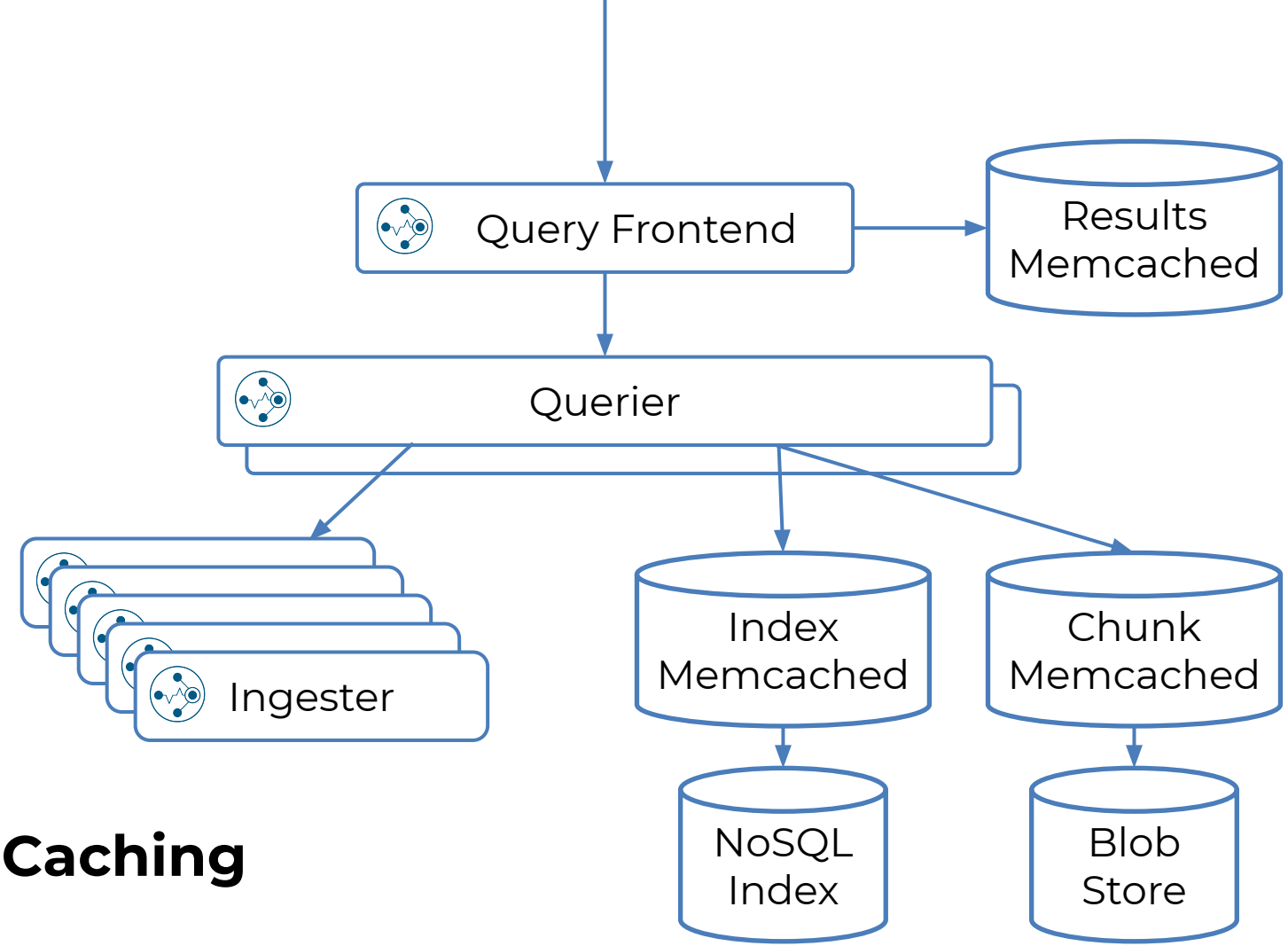


Merging / Deduping samples

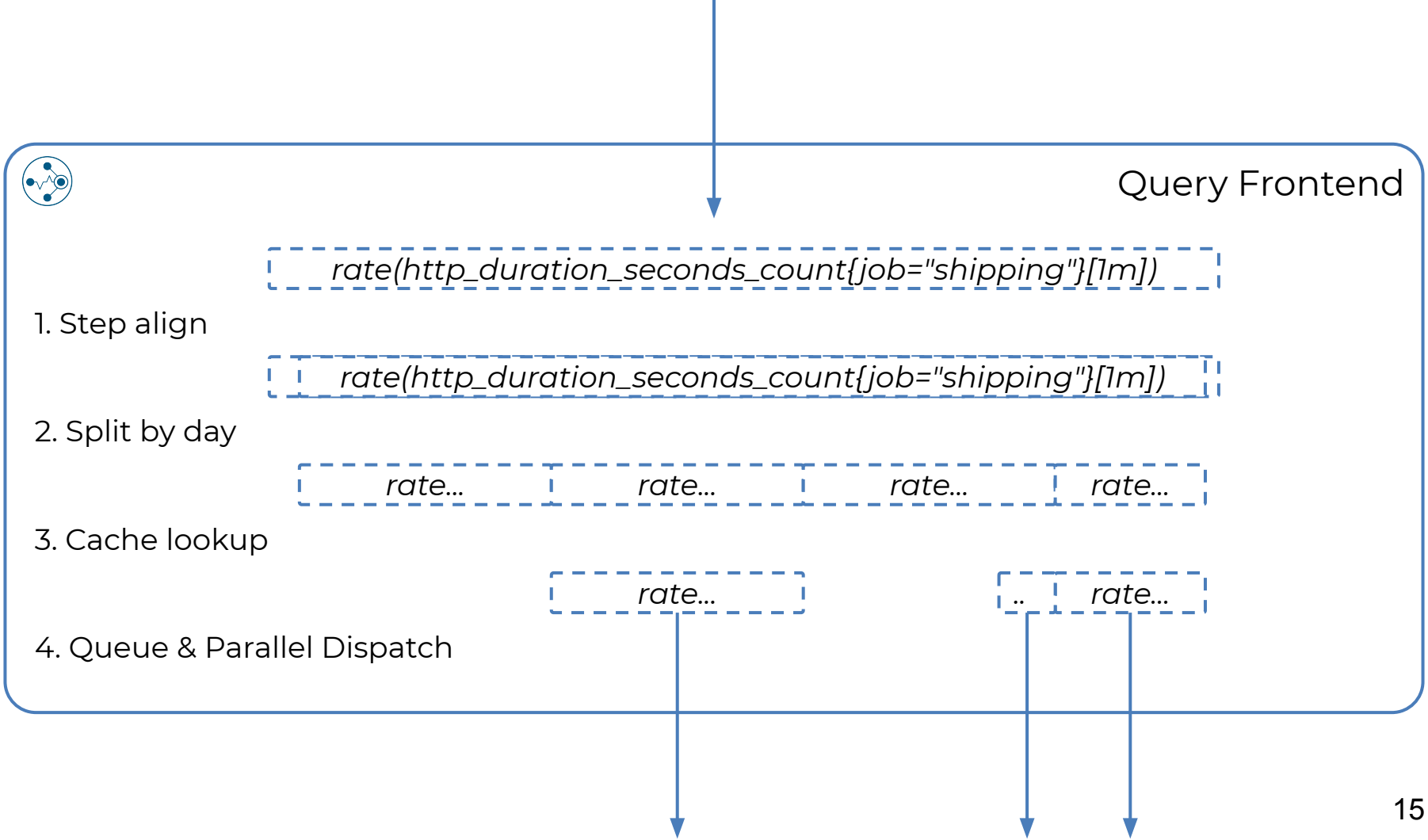
Method	CPU	Mem
Slices: Convert compressed chunks into []Samples; use recursive merge & dedupe on big slices.	**Fast**	**Huge**
Iterators: Iterate over chunks sample-by-sample, using a heap to merge and dedupe as we go.	**Slow**	**Tiny**
Identify **non-overlapping** sets of chunks, and build a non-deduping iterator for these.	**~Slow**	**Tiny**
Batches: Iterate over chunks a batch at a time. Detect non-overlapping chunks. Merge using heap.	**Fast**	**Tiny**


Querier
PromQL Engine
Ingester Client
Chunk Store
Ingester
Index
Memcached
Chunk
Memcached
NoSQL
Index
Blob
Store

Caching



More Caching


Query Frontend
rate(http_duration_seconds_count{job="shipping"}[1m])
1. Step align
rate(http_duration_seconds_count{job="shipping"}[1m])
2. Split by day
rate...
rate...
rate...
rate...
3. Cache lookup
rate...
..
rate...
4. Queue & Parallel Dispatch

So.. How fast?

<50ms (avg)

<500ms (P99)

Across Grafana Labs production clusters on week ending 14th May.

Thank You!

https://github.com/cortexproject/cortex