Google

Monarch

Google’s planet-scale streaming monitoring infrastructure.

Background

Architecture and Data Model

Queries

Using Monarch

Monarch Platform

Lessons Learned re: Scaling

Monitoring at Google

Ref: https://www.google.com/about/datacenters/inside/locations/index.html

Monitoring at Google

Global Span

Huge Volume

Many Kinds

- Hardware/networking
- OS
- Infrastructure services
- Big, user-facing services
- Smaller services

Constant change

Ref: https://www.google.com/about/datacenters/inside/locations/index.html

Essentials of Monarch Scaling

Maintain good hygiene
Scale horizontally
Reduce dimensions early

Background

Architecture and Data Model

Queries

Using Monarch

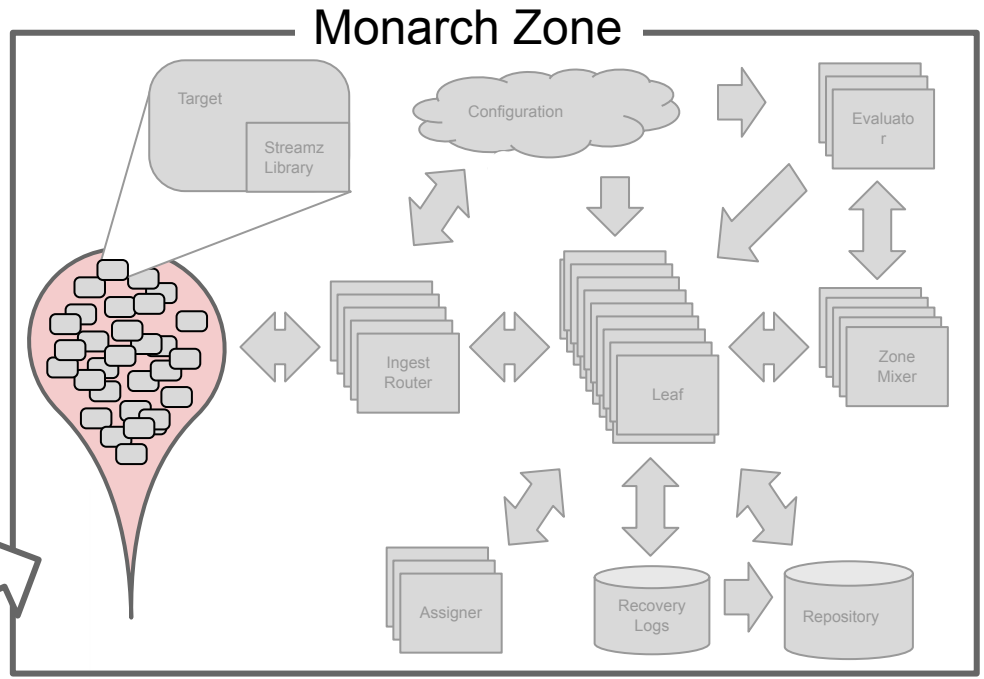
Monarch Platform

Lessons Learned re: Scaling

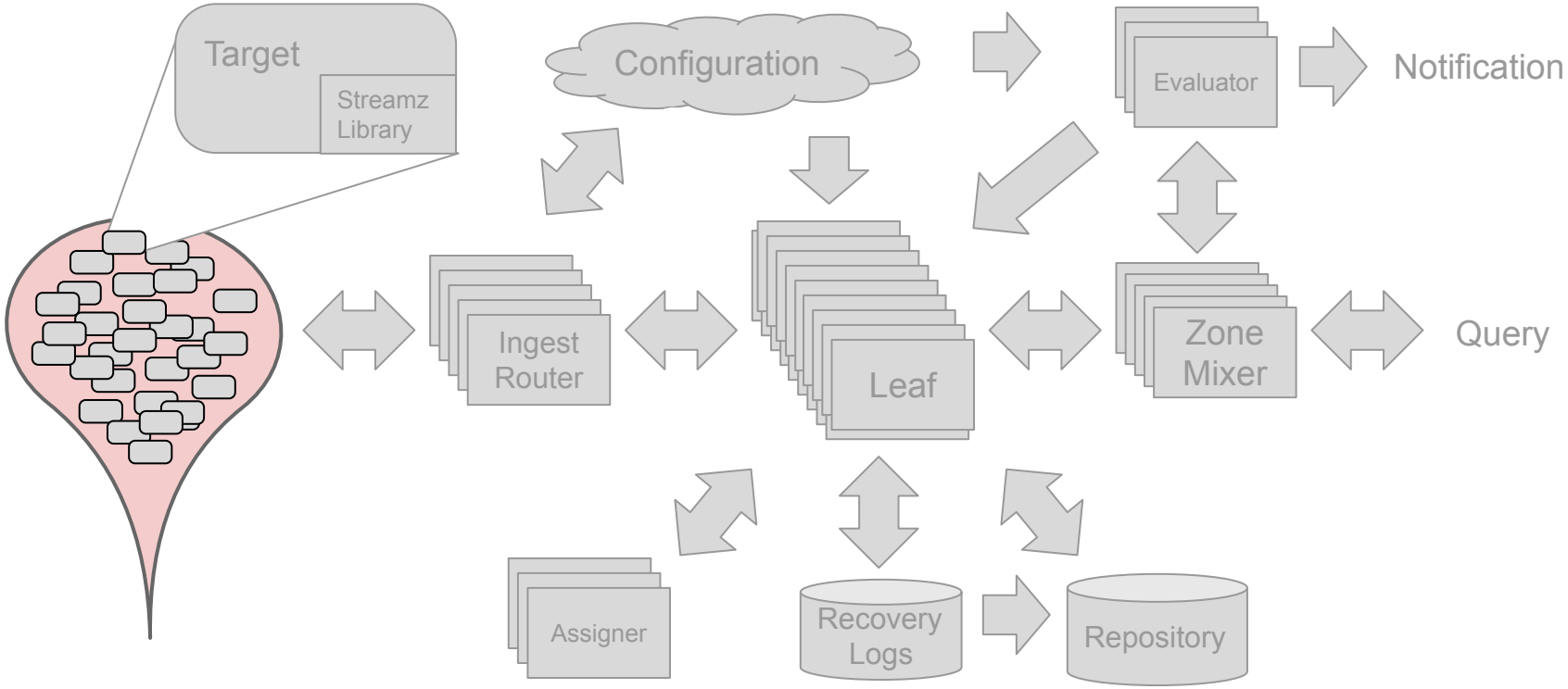
Global Extent

Ref: https://www.google.com/about/datacenters/inside/locations/index.html

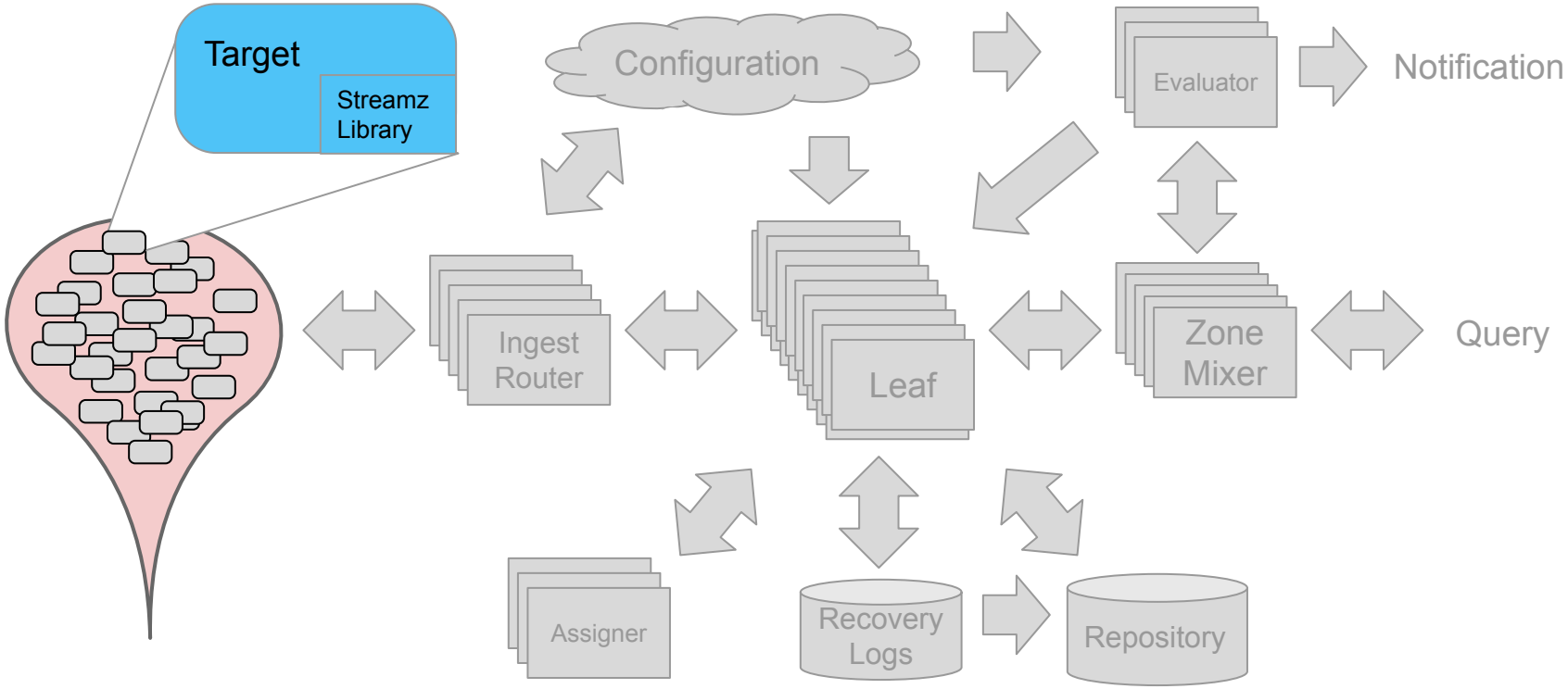
Monitor Locally



Monarch Zone: Ingestion, Retention and Queries



Monarch Zone: Ingestion



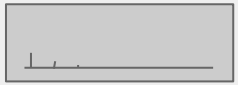
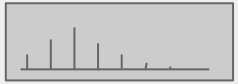
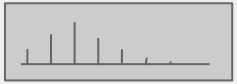
Metrics

Description

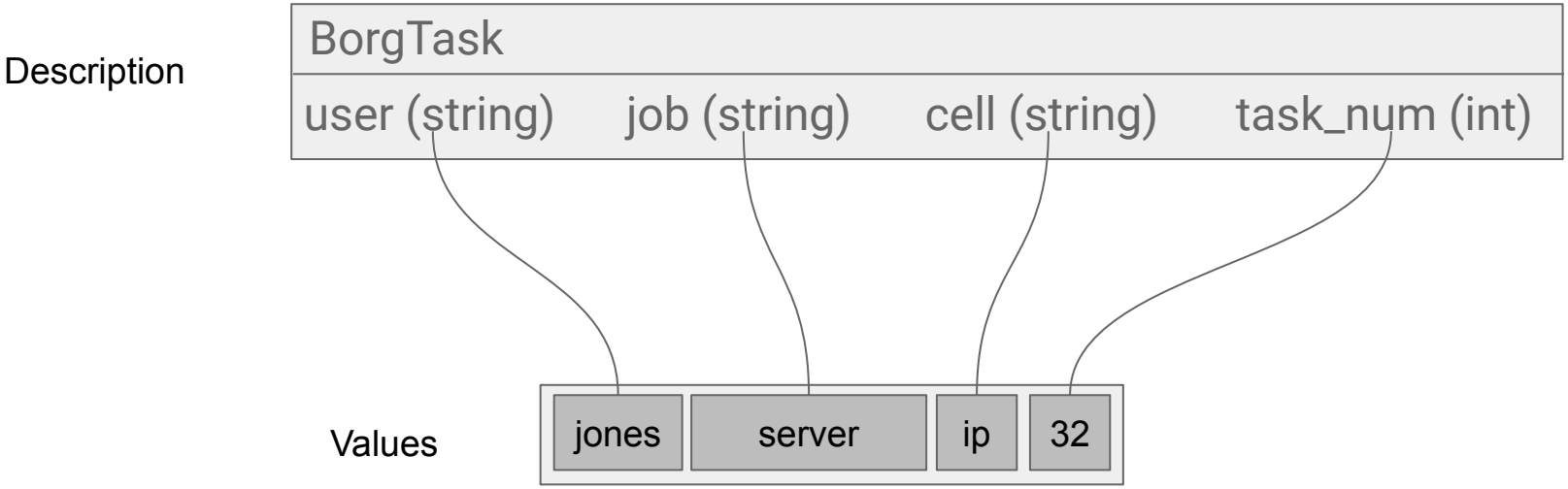
/http/server/response_latencies

Path (string) Status_code_class (int64) (Distribution) (cumulative)

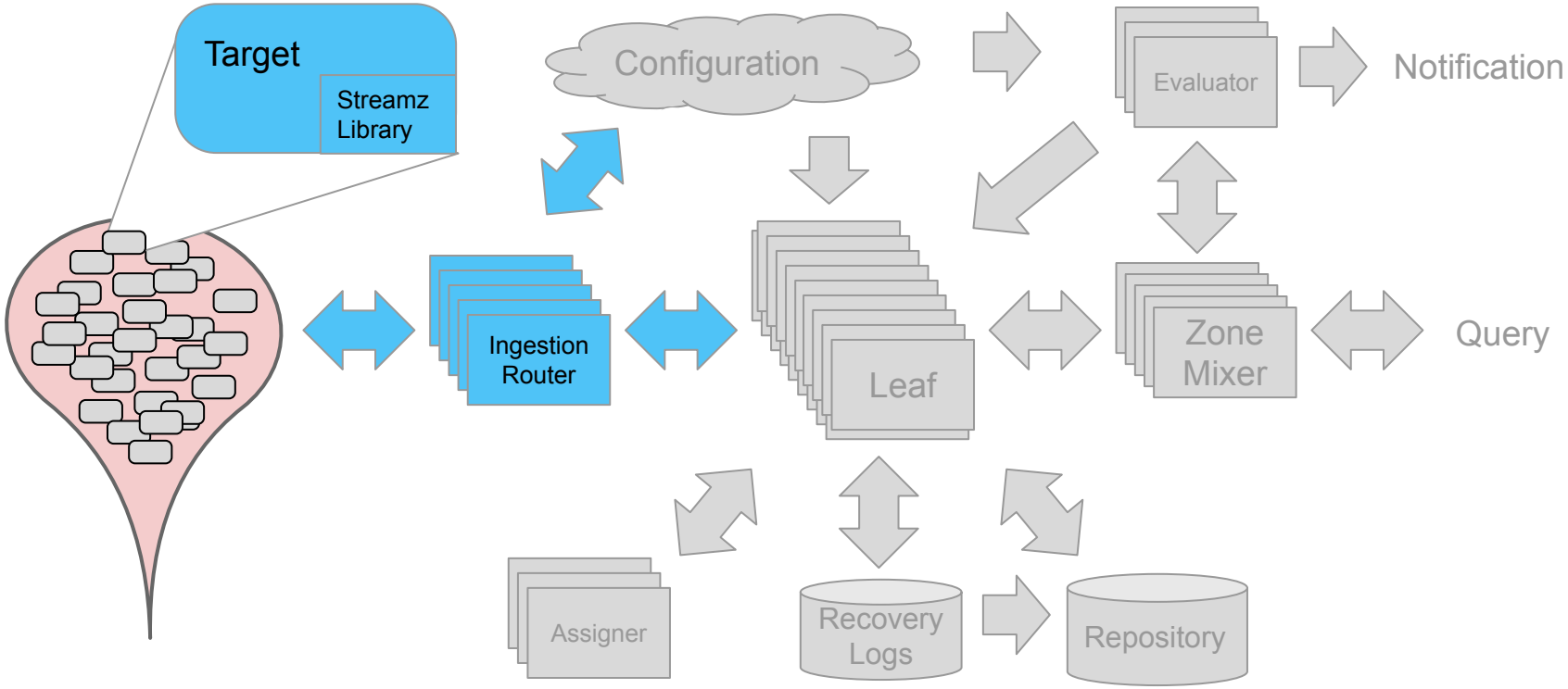
Values

/requestz	200	
/requestz	500	
/inspectz	200	
/statusz	200	
...		...

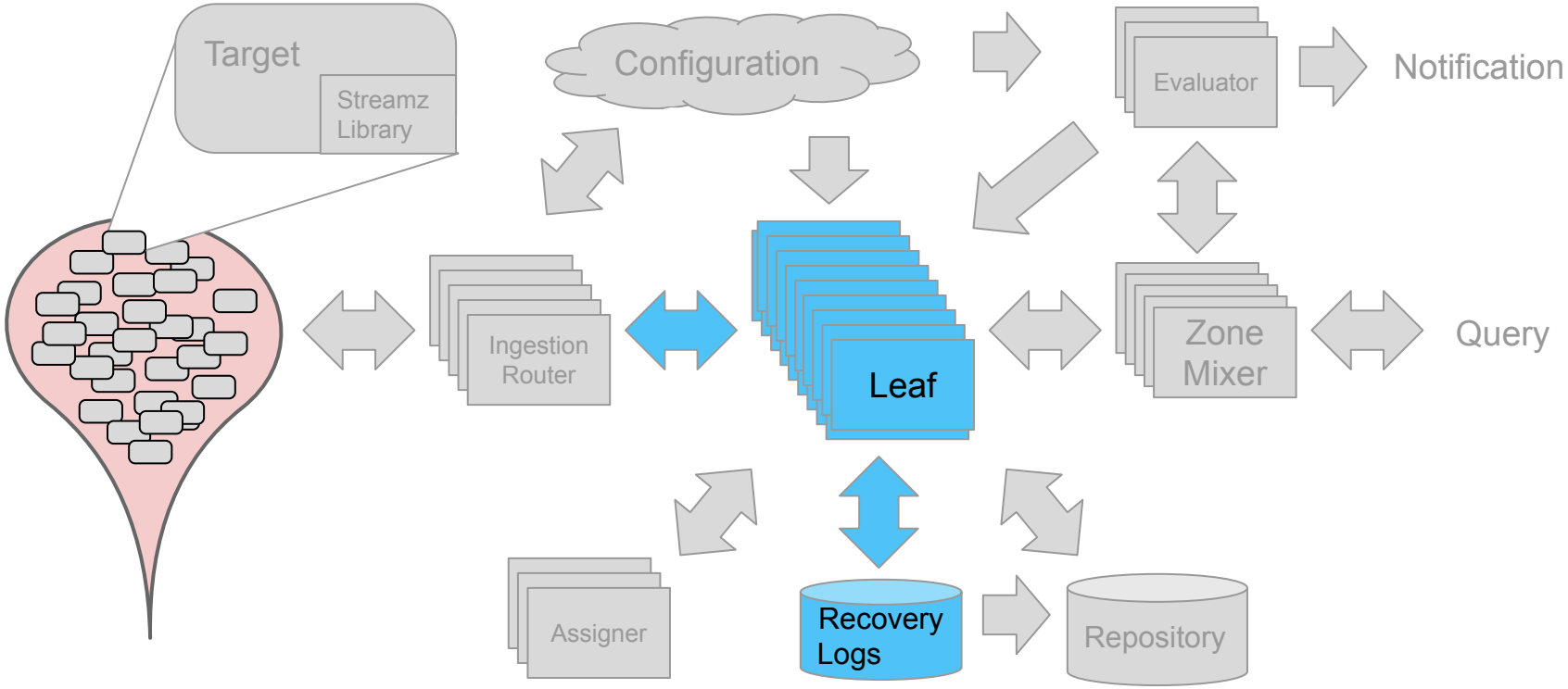
Target Schema



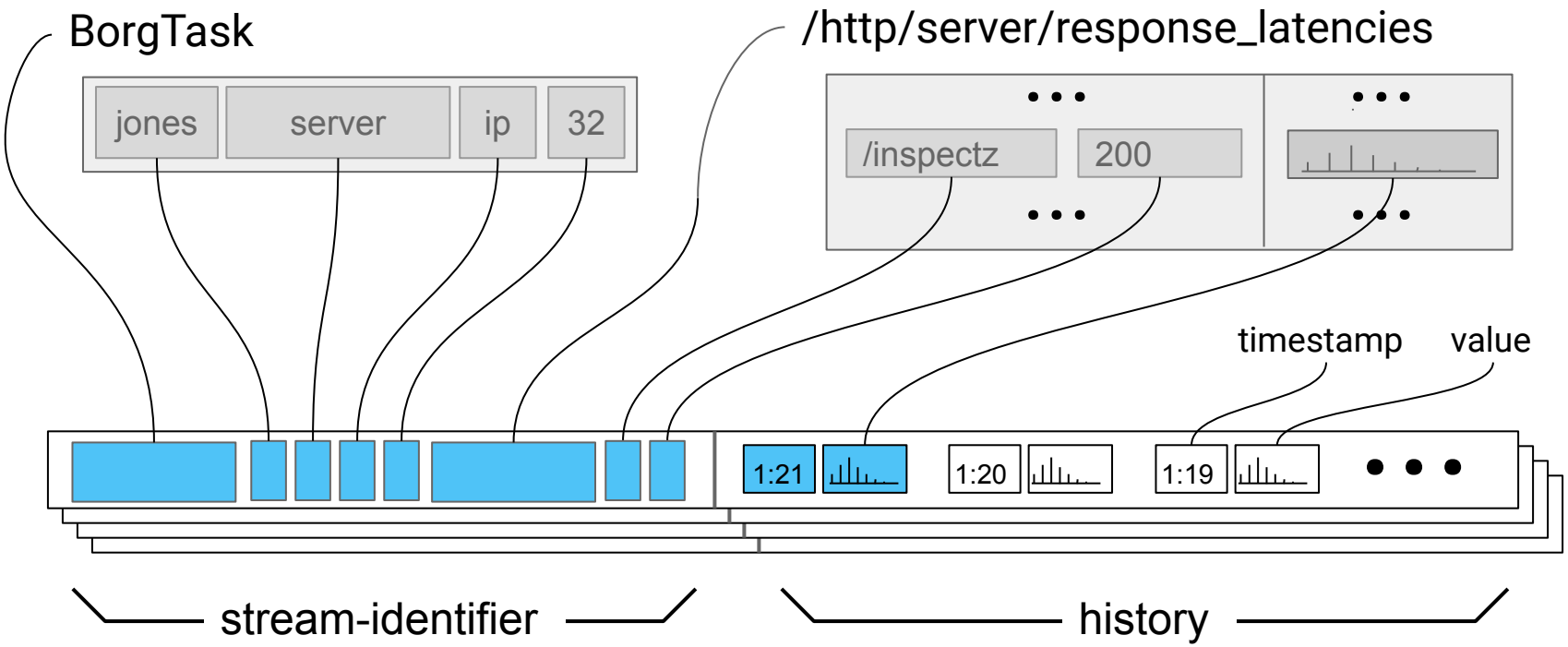
Monarch Zone: Ingestion



Monarch Zone: Retention



Streams


BorgTask
jones
server
ip
32
/http/server/response_latencies
/inspectz
200
timestamp
value
1:21
1:20
1:19
stream-identifier
history

The Data Model for Queries

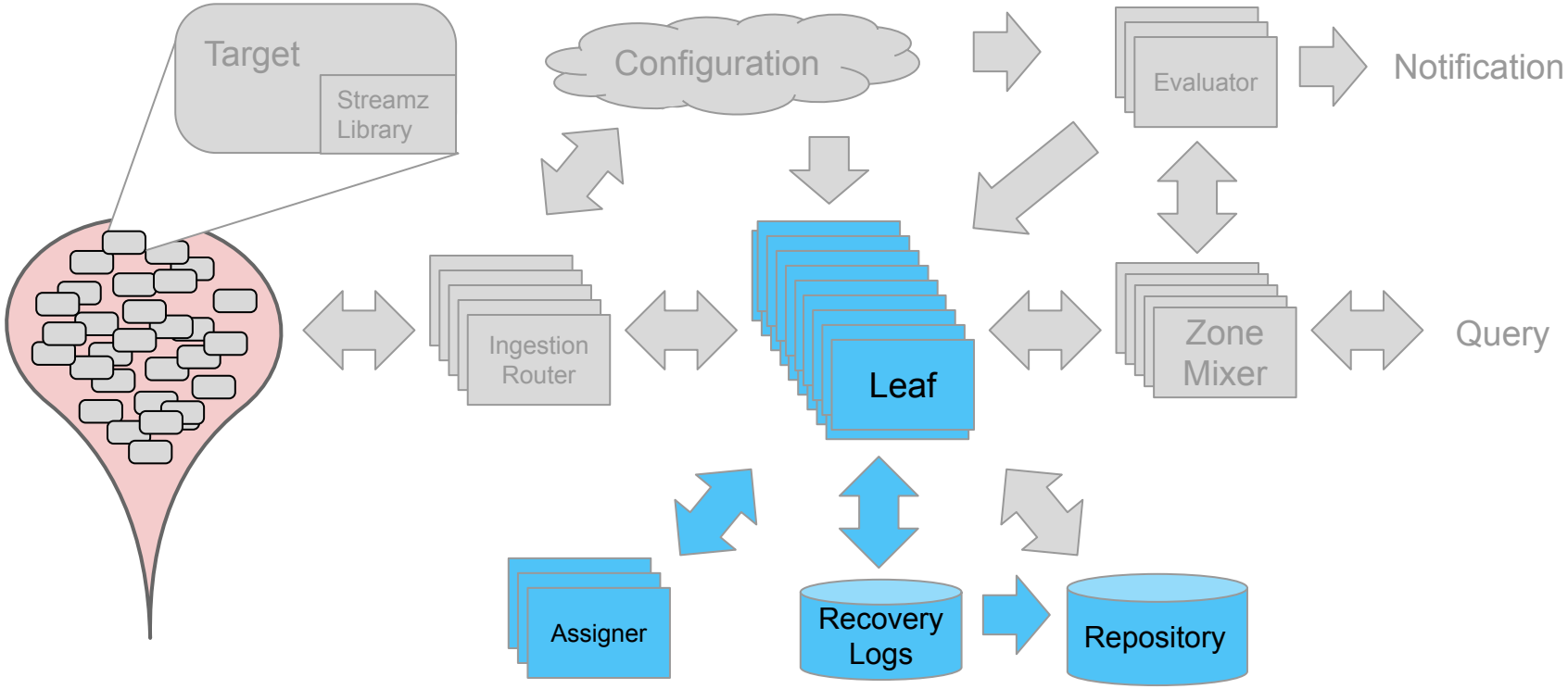
BorgTask :: /rpc/server/server_latencies

user	job	cell	task_num	status_code_class	path	server_latencies
jones	server	ip	0	DB	Alloc	10:52-1:21 10:42-01:21 ...
.	.	.	.	.	.	
.	.	.	.	.	.	
.	.	.	.	.	.	
jones	server	ip	876	DB	Query	10:52-1:21 10:42-01:21 ...
jones	server	ip	877	DB	Undo	10:52-1:21 10:42-01:21 ...
...	...	...	...	...	...	
emons	client	qr	33	Help	Ask	07:33-4:49 07:38-4:49 ...

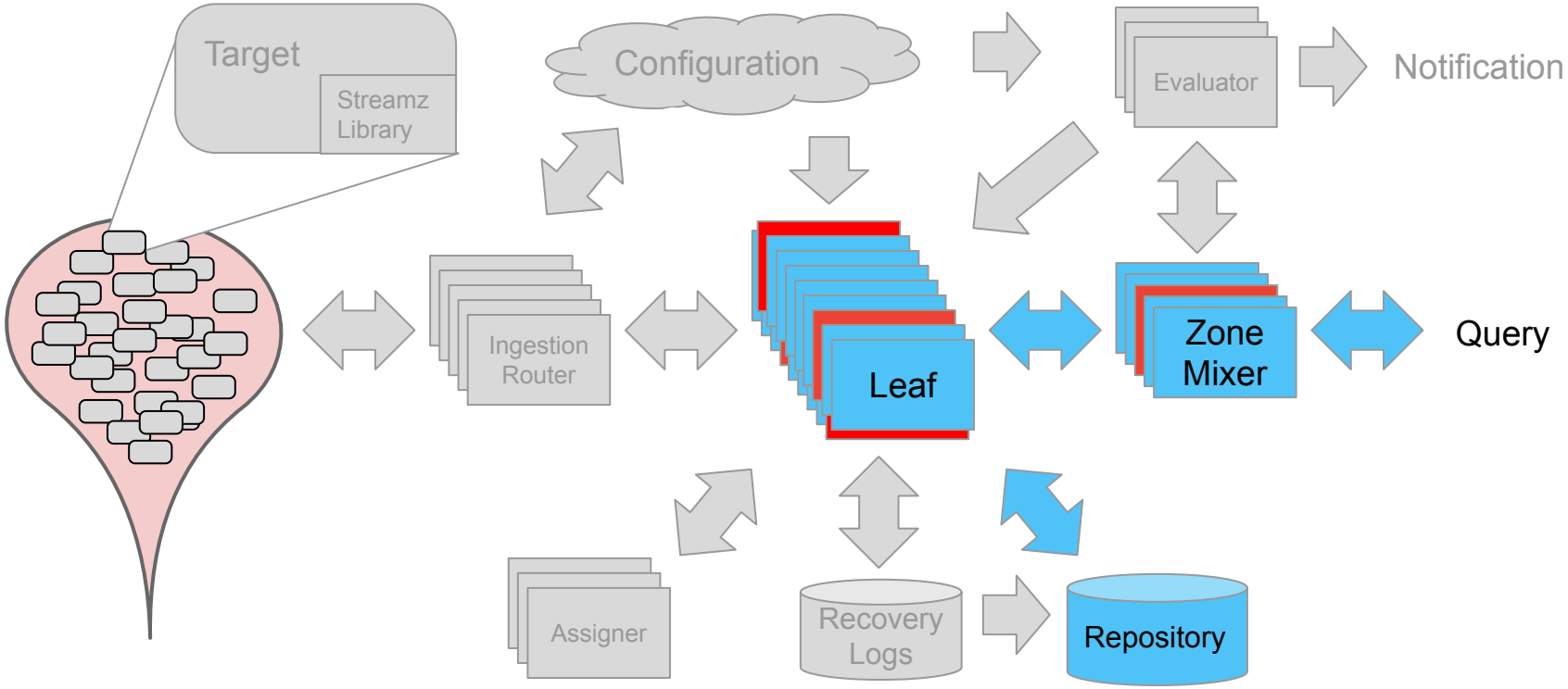
stream-id columns

time series column

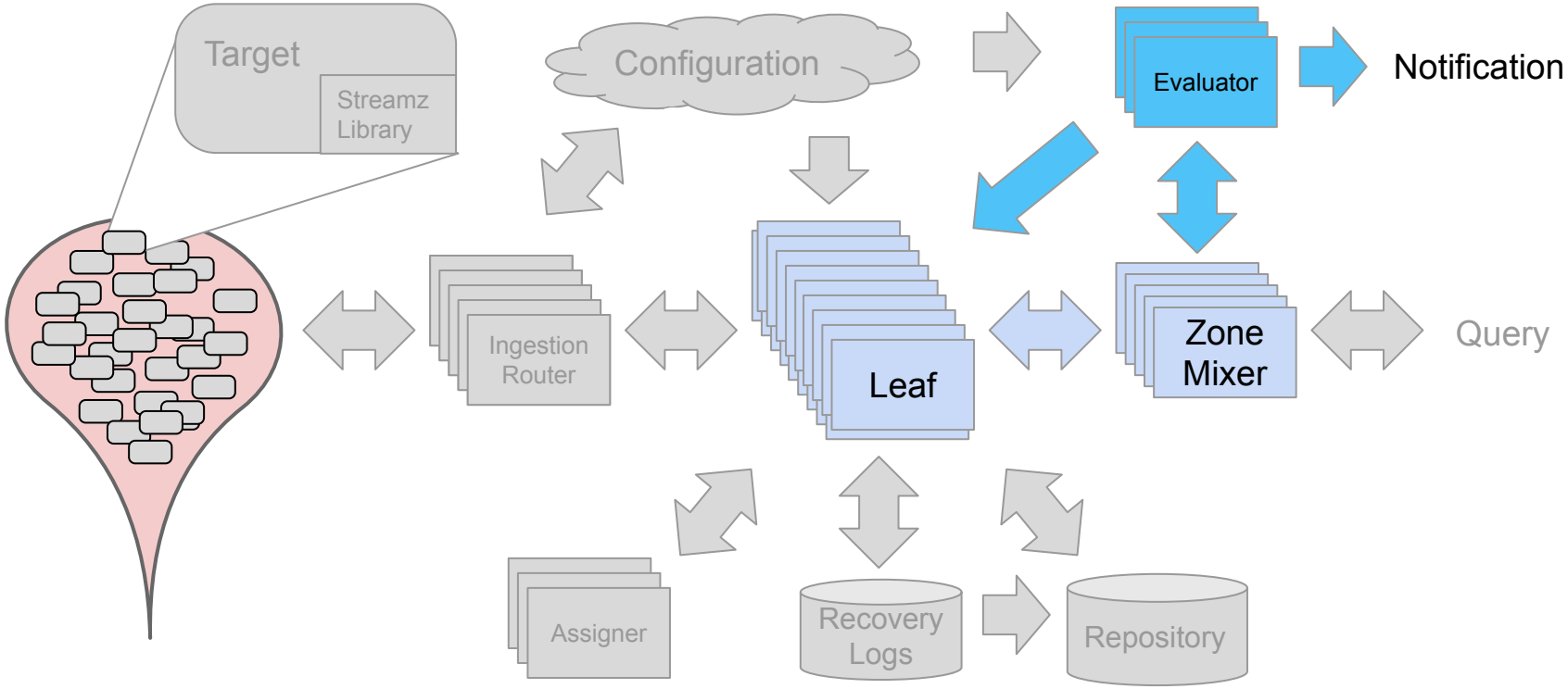
Monarch Zone: Retention



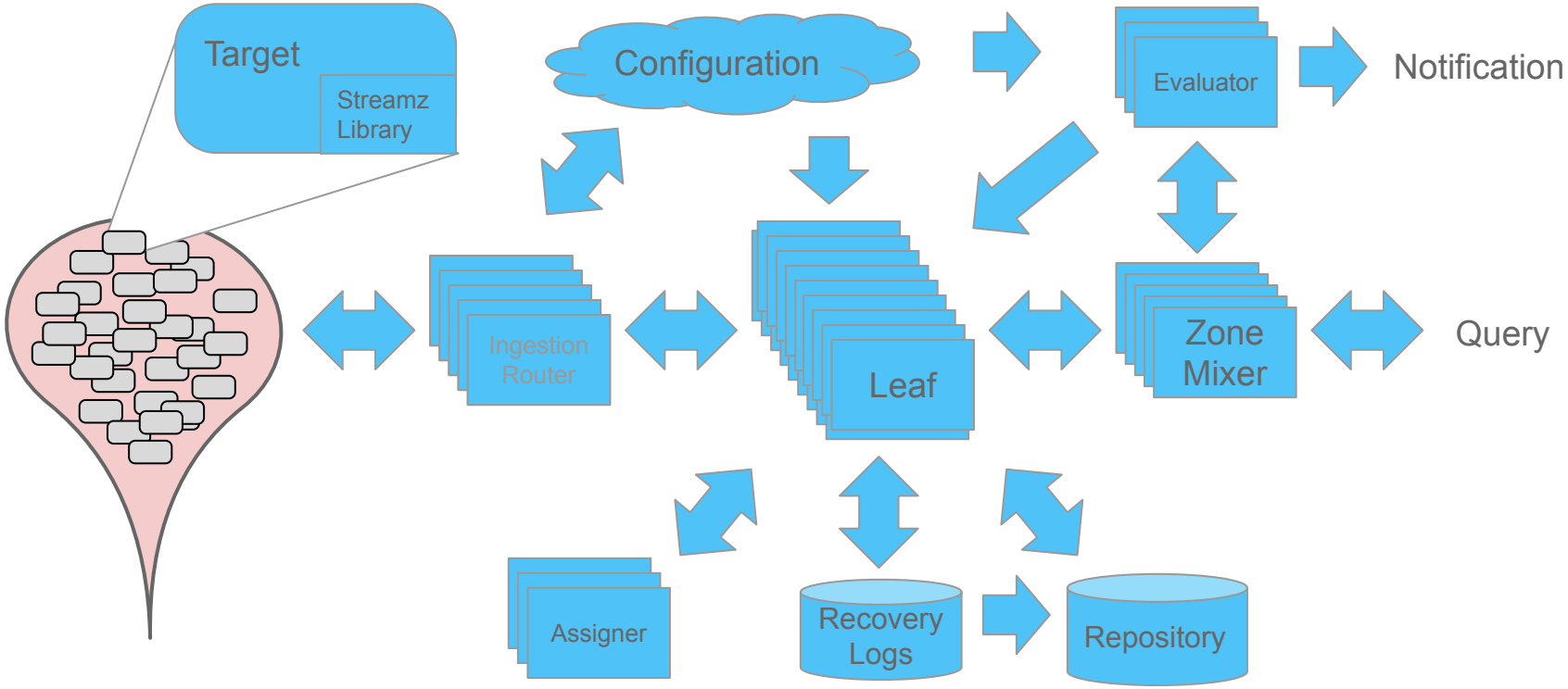
Monarch Zone: Query



Monarch Zone : Evaluation and Notification

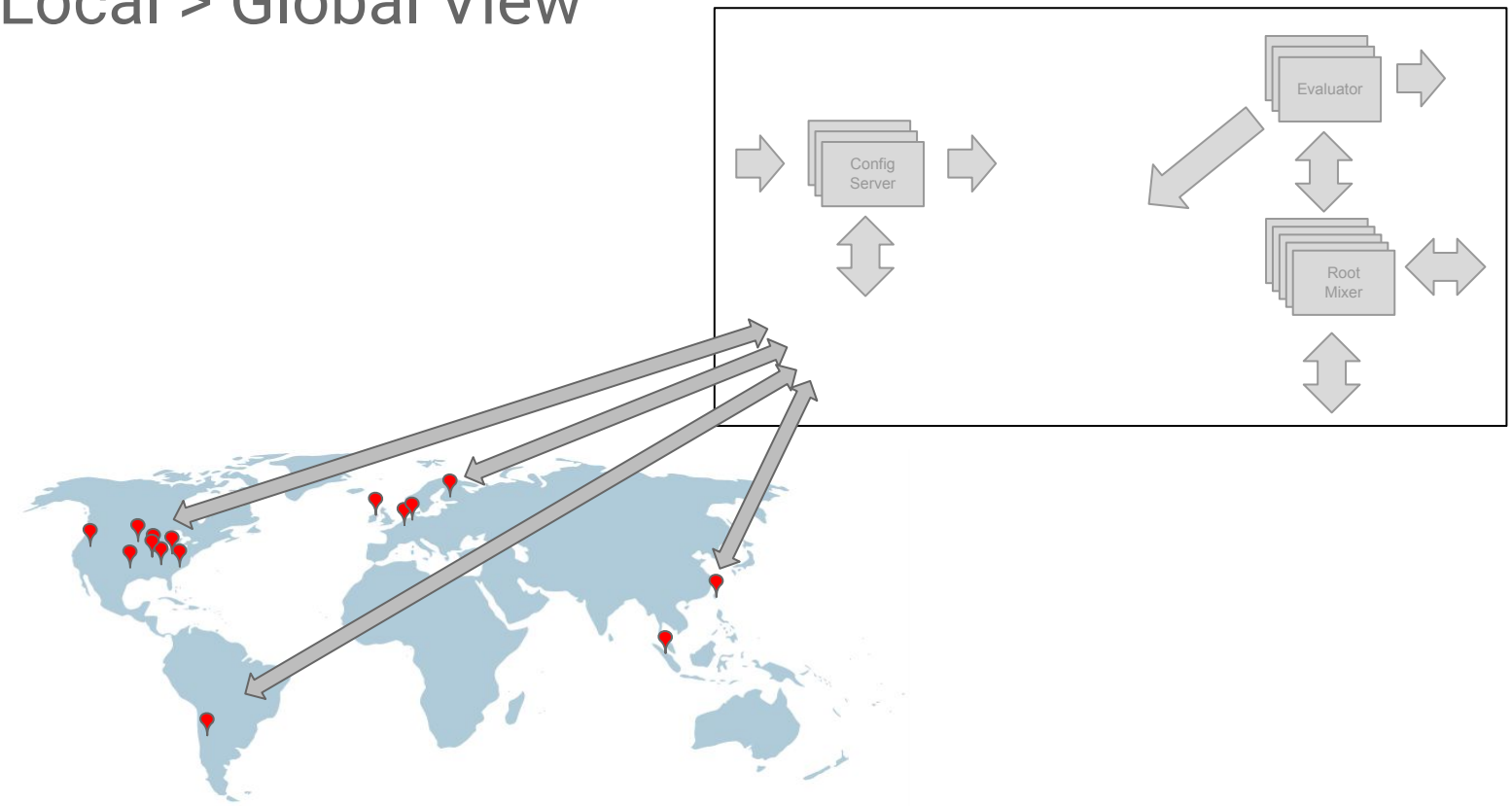


Monarch Zone

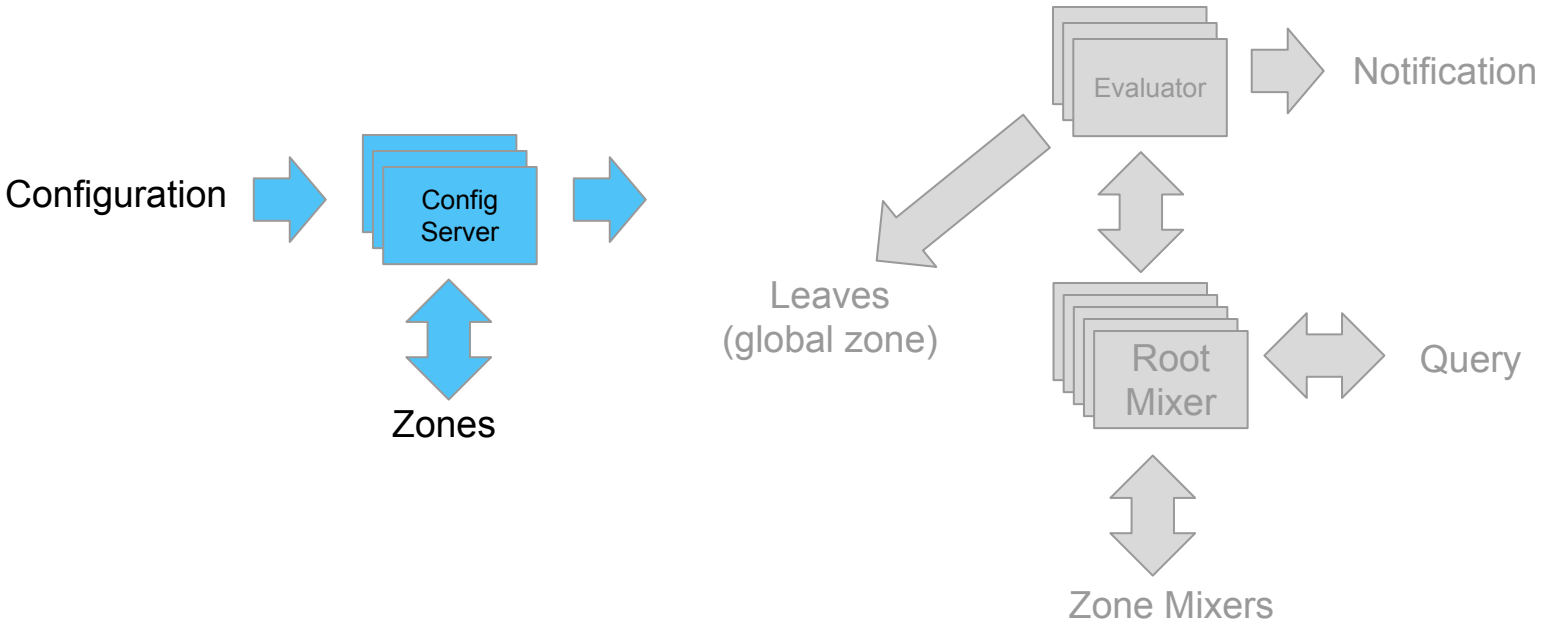


Ref: https://www.google.com/about/datacenters/inside/locations/index.html

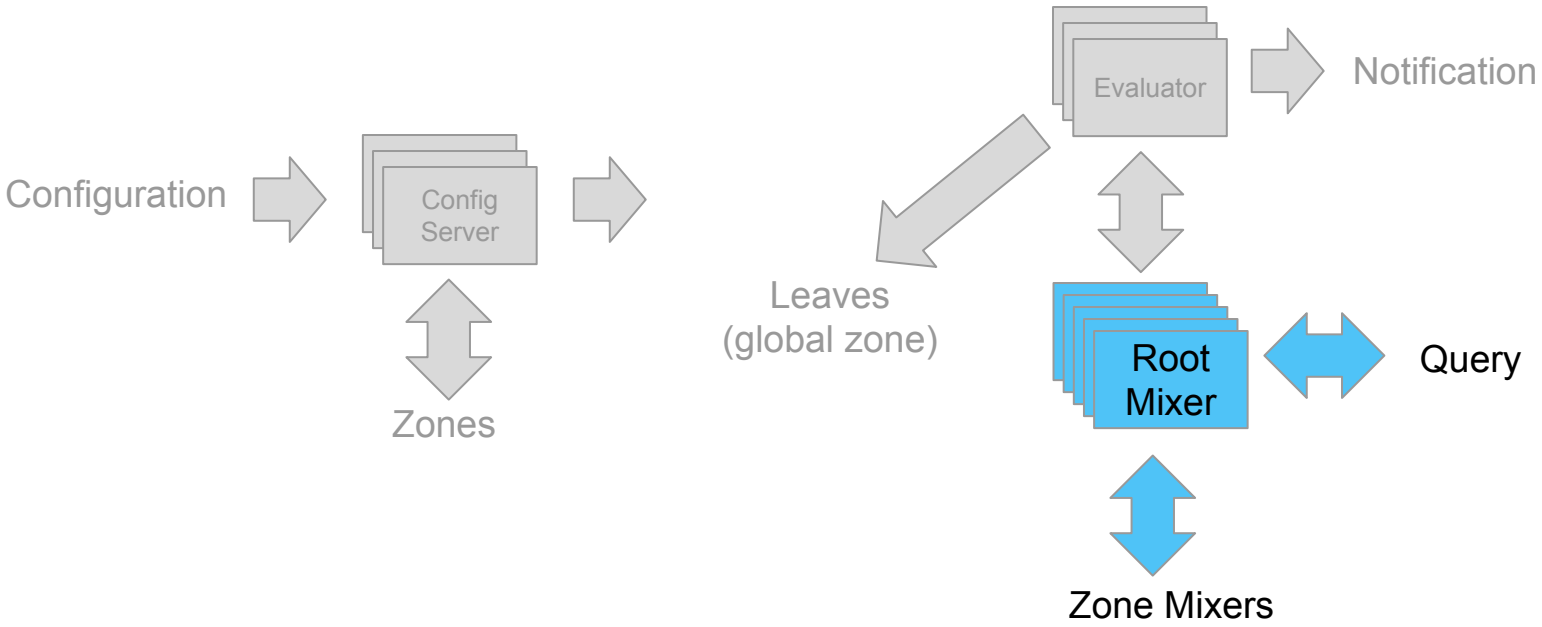
Local > Global View



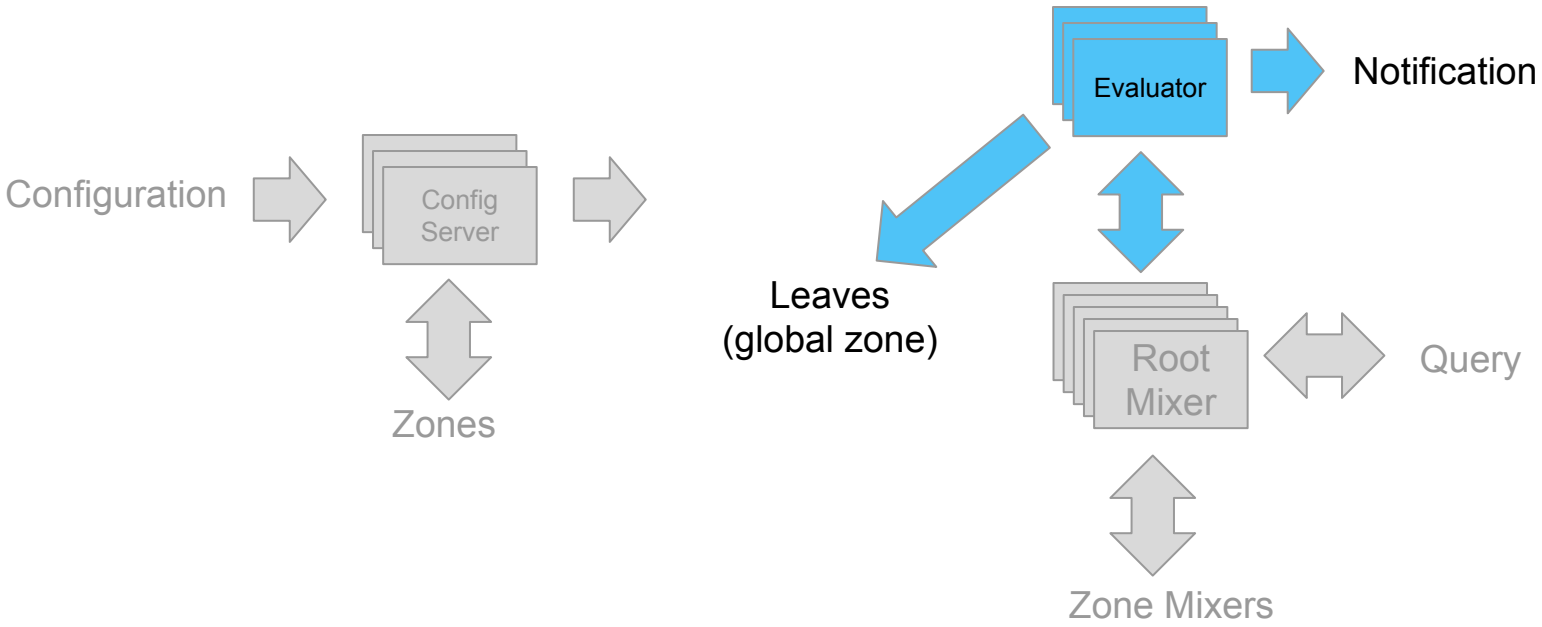
Global Monarch



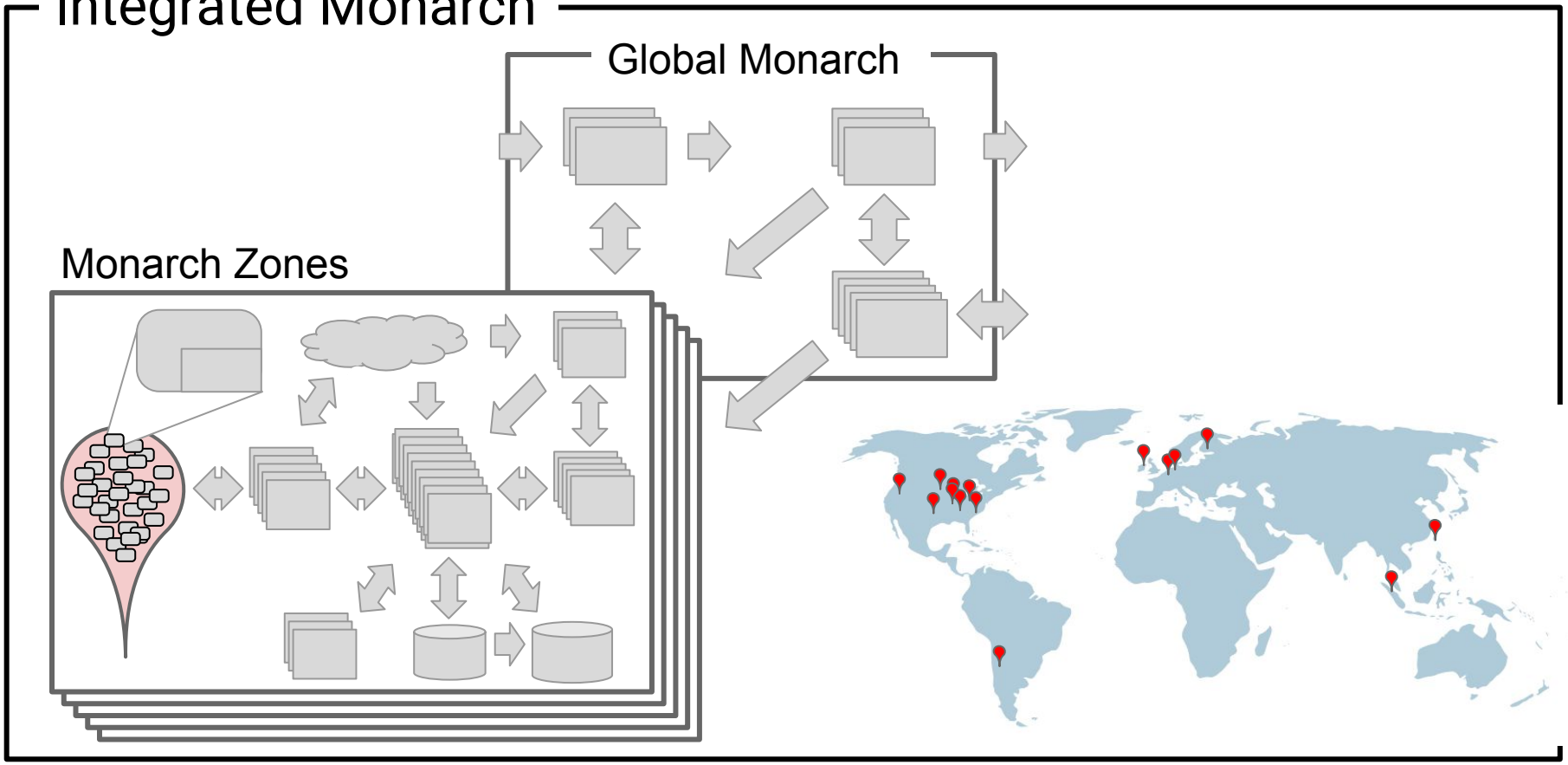
Global Monarch



Global Monarch



Integrated Monarch



Background

Architecture and Data Model

Queries

Using Monarch

Monarch Platform

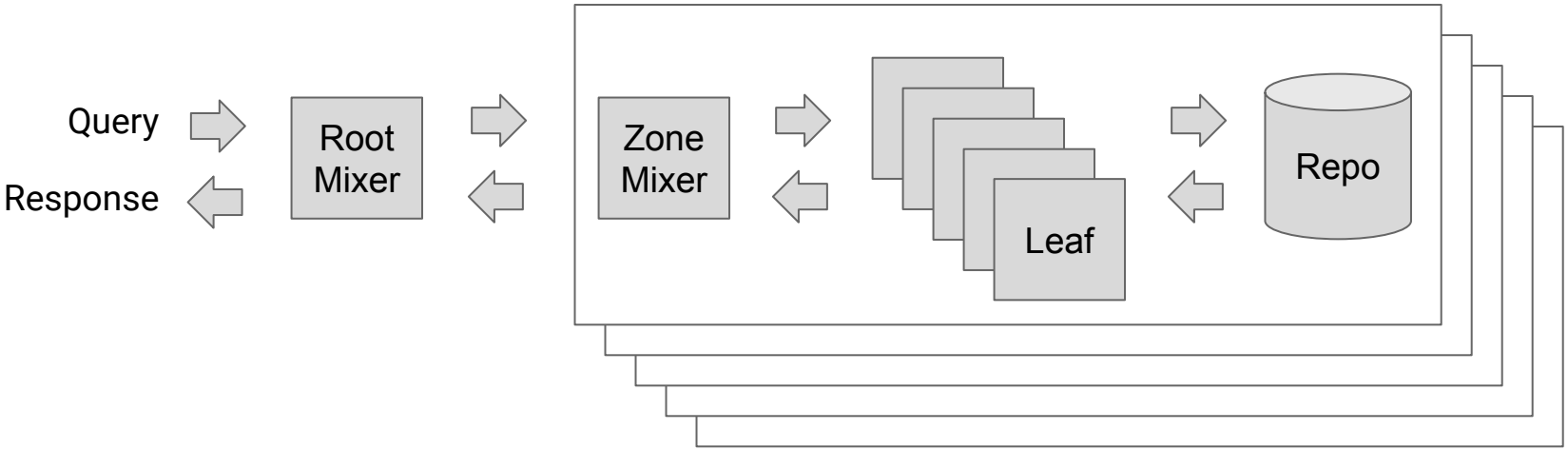
Lessons Learned re: Scaling

Query

```
Query(
  Fetch(Raw('BorgTask', '/http/server/response_latency'),
            {'user': 'gmail', 'status_code_class': 200}) |
  Window(Delta('5m')) |
  GroupBy([job, cell], Sum()) |
  Point(Percentile(95)), '1h', '5m')
```

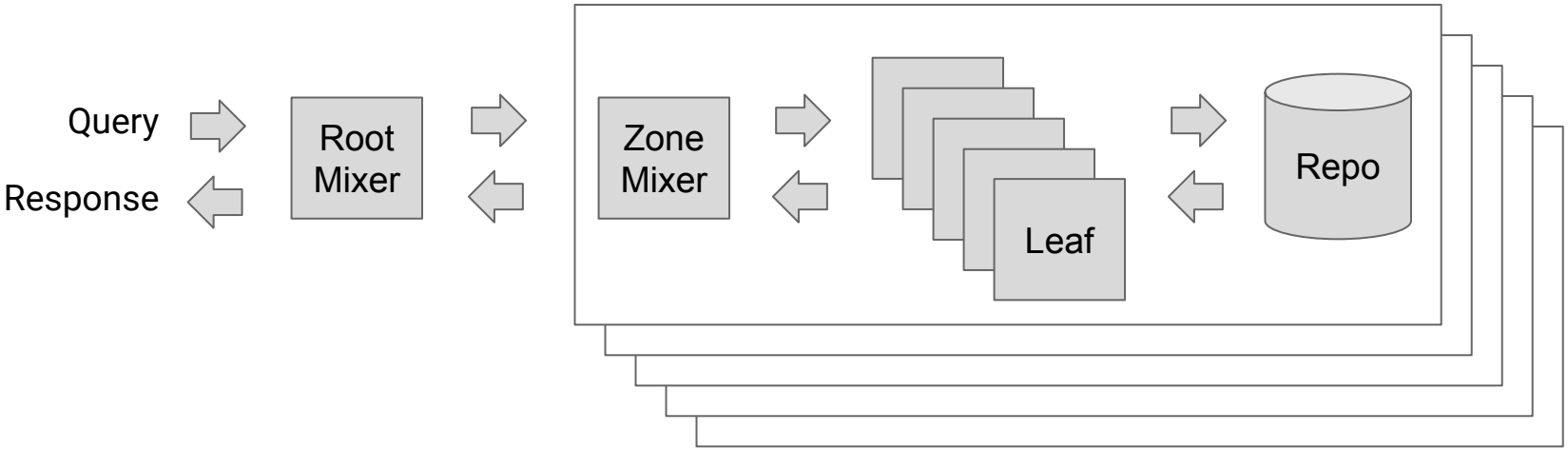
Also: `Join, PickTopStreams, MapStreamId, Union`
General expressions
A large set of aggregation functions

The Life of a Query



Fetch
Window
GroupBy
Point

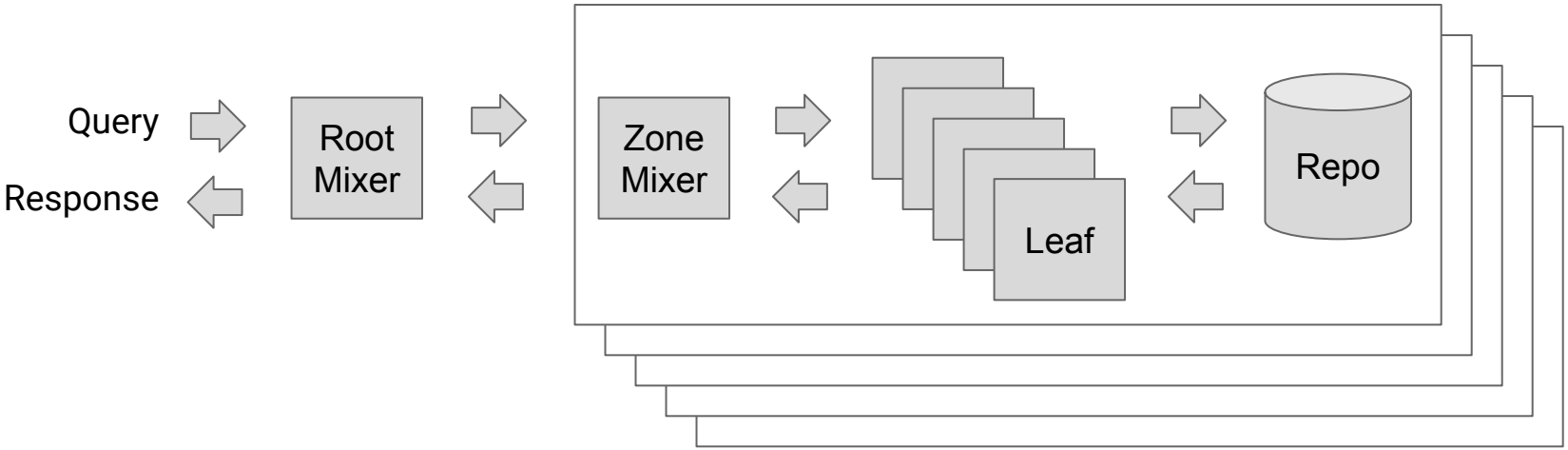
The Life of a Query



Fetch
Window
GroupBy
Point

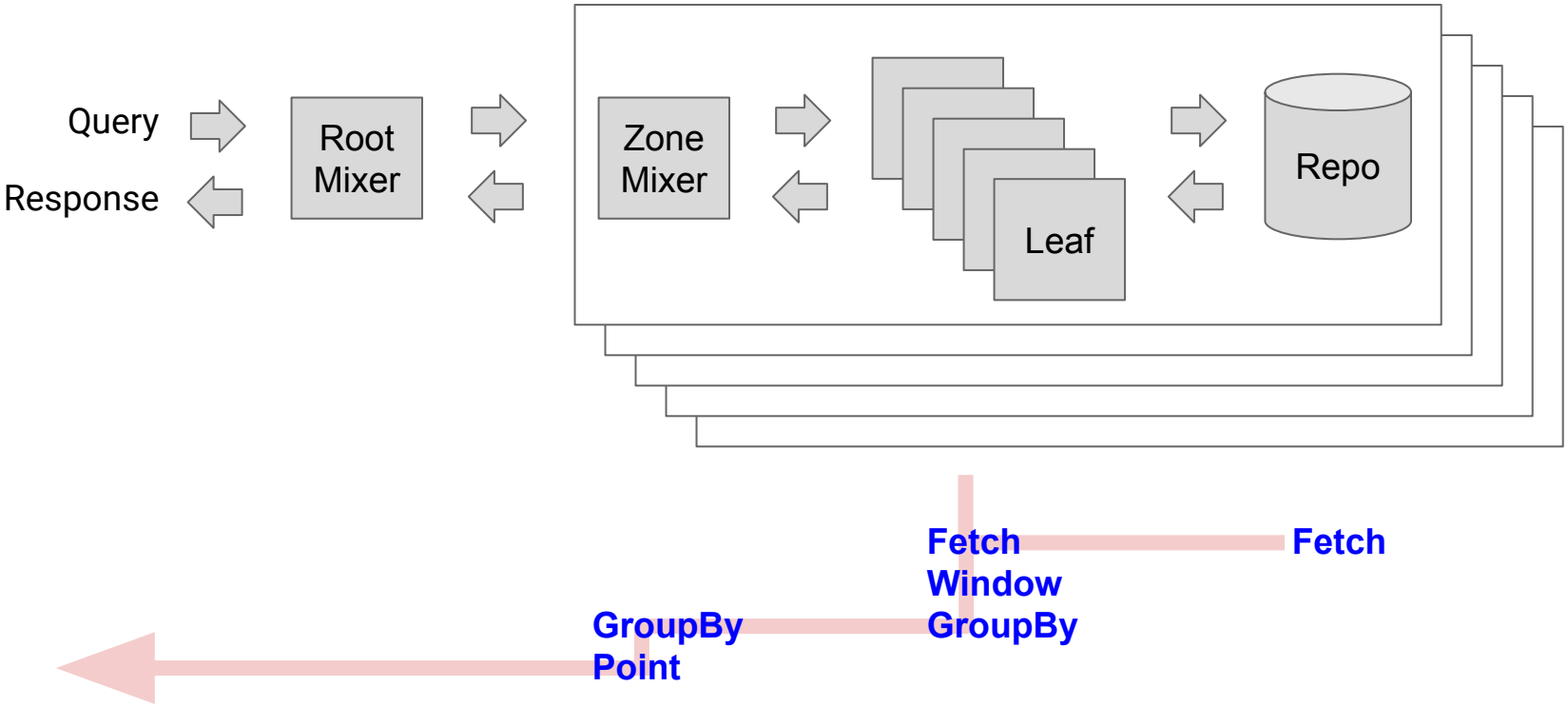
Fetch
Window
GroupBy
Point

The Life of a Query



Fetch	Fetch	Fetch	Fetch
Window	Window	Window	
GroupBy	GroupBy	GroupBy	
Point	Point		

The Life of a Query



Background

Architecture and Data Model

Queries

Using Monarch

Monarch Platform

Lessons Learned re: Scaling

Panopticon

Home Query playground Select console Settings jbanning@ jbanning Change role Help

Panopticon

CONSOLE SUBPAGES	PANELS	QUERIES	RECENT USAGE DESCRIPTIONS
QueryEngineStats Page	10 panels	14 queries	last edit by: jbanning on: 2015/06/03 09:06:41 (no description)
test Page	6 panels	6 queries	last edit by: jbanning on: 2013/12/10 11:52:29 (no description)

https://pcon.corp.google.com/p#jbanning/QueryEngineStats/Page

CREATE NEW CONSOLE

A console, or dashboard, is a designated space to organize and view your data. You can use them to render graphs or tables from saved queries.

CONFIGURE RETENTION POLICIES

Retention policies decide what data is to be collected and from which jobs. They also specify how often to collect the data, what medium to store it in, and how long it should be stored for.

MANAGE QUERIES

Queries range from simple to complex sets of predicates that specify what data to pull from Monarch. They are also the building blocks to create alerts as well as graphs and tables on your consoles.

CONFIGURE ALERT DESTINATIONS

This is where you designate who, or what, gets paged when something is on fire. You can also configure escalation rules.

Using Panopticon

Retention Policy

Home Query playground Select console Settings jbanning@ jbanning Change role Help

Queries

Collection

Panopticon

+ Add Storage Policy

15m@1s

1w@5m

8d@30s

+ Add Collection Path

/monarch/resource_manager/memory_managed

/monarch/resource_manager/memory_reserved

/monarch/resource_manager/memory_used

/monarch/resource_manager/num_big_clients

/monarch/resource_manager/num_clients

/monarch/resource_manager/num_waiting_big_clients

/monarch/resource_manager/num_waiting_clients

Usage

Alert Destinations

Revisions

1w@5m Edit Delete

@5m for 1w in memory

@5m for 8w in bigtable

Traffic class: BE1

Replication level: 1

This policy is not covered by our SLA

Collection path	Collection target
/monarch/resource_manager/memory_managed	**monarch.BackendTask** monarch_namespace=auto monarch_job=mixer

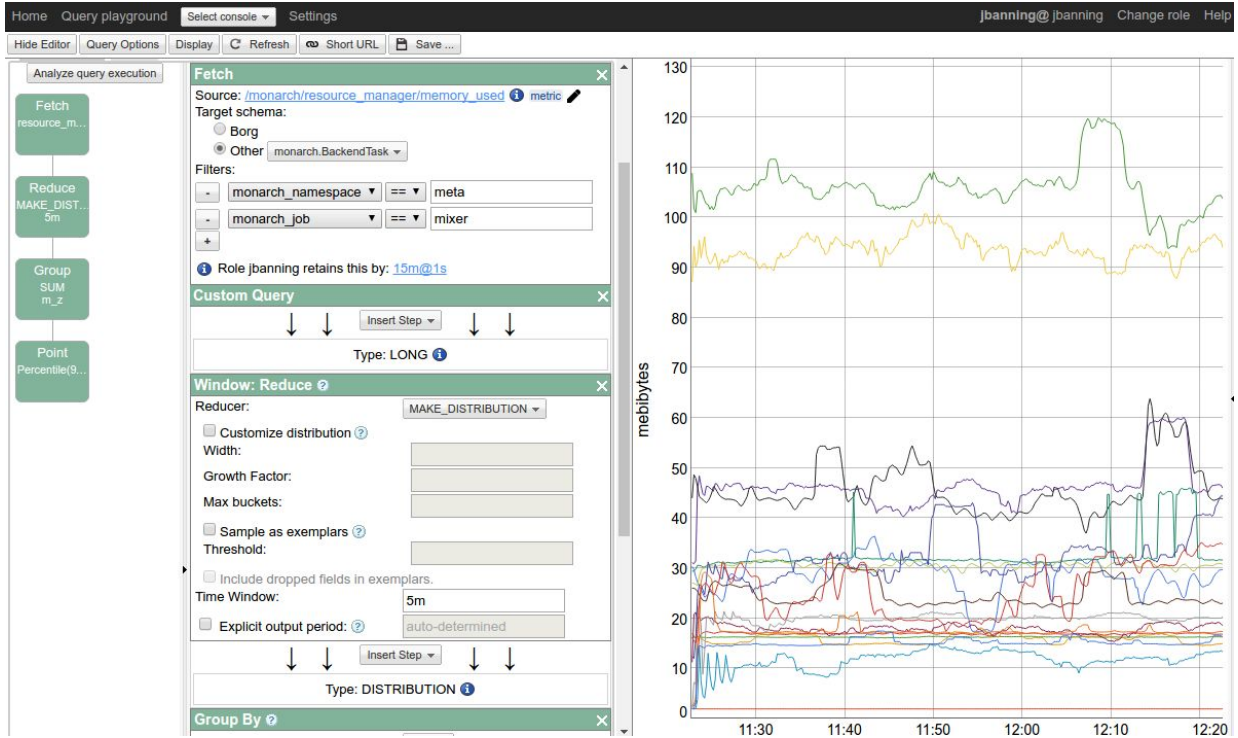
Precomputed('jbanning', '__panopticon', 'Distribution of Auto Memory Used over 5m')
from query: Memory Used Auto By Cell and Job

Precomputed('jbanning', '__panopticon', 'Resident Queries by Cell 5m')
from query: Resident Queries By Cell and Job

Using Panopticon

Retention Policy

Query



Using Panopticon

Retention Policy

Query

Configure alert

Home Query playground Select console Settings jbanning@ jbanning Change role Help

Queries

Panopticon
+ Add Query
95th % CPU Per Query (ms)
99% Memory Used Mixers
99th % CPU Per Query (ms)
count > 64
Engine 1m QueryRate from /monarch/engine
Engine Big Query 1m Rate
Engine Big Query 1m Rate -- direct
Engine Query 1m Rate
Engine Query 1m Rate -- direct
greater than growing
machine states
machine states (copy)
Mean CPU Per Query (ms)
Memory Reserved Mixers
Memory Used Auto By Cell and Job
Memory Used Mixers
Min Memory Managed Mixers
Num Resident Queries Mixer (1h)
Num Waiting Clients Auto Mixer - raw
Query Memory Used Auto Mixer (1h)
Resident Queries By Cell and Job
sec_per_sec

Collection
Usage
Alert Destinations
Revisions

Editing 'Memory Used Auto By Cell and Job'

Insert Step

Type: DOUBLE

Collection and Retention

Query is retained (No further action required)

Create rule for the raw data fetched by this query in retention policy:

Precomputations and Alerts

Store as precomputed data

Name:

Distribution of Auto Memory Used over 5m

Evaluation and retention: 1w@5m

This precomputation will be evaluated every 5m

Create an alert

Name:

Excess memory

Alert if value < 150 for 1h

Alert destination: EMAIL

Evaluation and retention: 1w@5m

This alert will be evaluated every 5m

Suppress alert notifications on drain

How far back in time the last impact should be: 2h

Job is supercluster-enabled

Specify custom impact types

Field containing borg cell: borg_cell

Done Cancel

borg_cell	percentile	Latest
ih	50%	1.000
ih	90%	1.800
ih	95%	1.900
ih	99.9%	1.998
qb	50%	1.000
qb	90%	1.800
qb	95%	1.900
qb	99.9%	1.998
jg	50%	1.000
jg	90%	1.800
jg	95%	1.900
jg	99.9%	1.998
ya	50%	46.324
ya	90%	117.848
ya	95%	157.867
ya	99.9%	612.645
yb	50%	44.219
yb	90%	118.838
yb	95%	168.625
yb	99.9%	2274.034
yi	50%	32.975

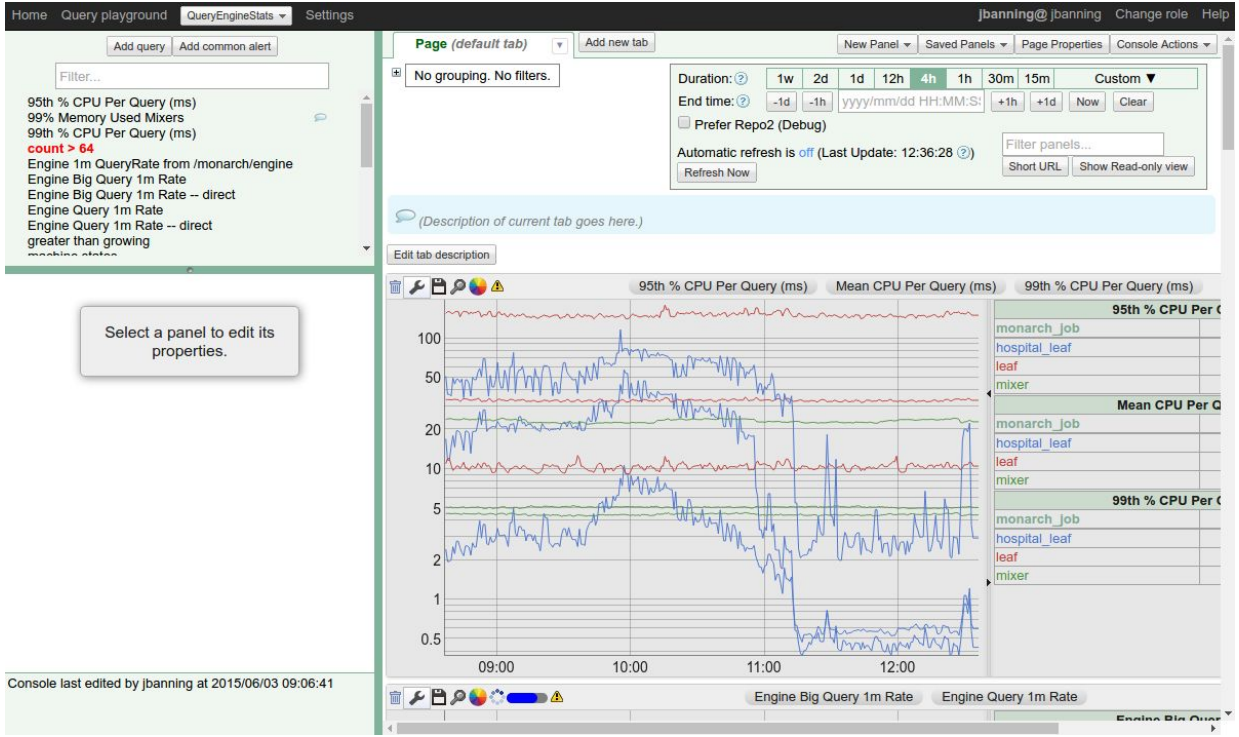
Using Panopticon

Retention Policy

Query

Configure alert

Setup Consoles



Google

Background

Architecture and Data Model

Queries

Using Monarch

Monarch Platform

Lessons Learned re: Scaling

Monarch as Platform

A custom console service

Python-based configuration libraries that encode best practices

Really automatic monitoring

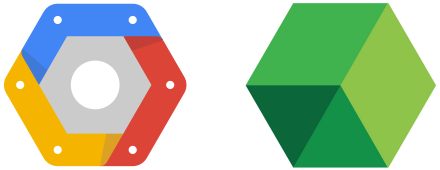
Cross company monitoring

SLA definition and alerting

Automated monitoring of rollouts

• • •

Google Stackdriver



Monarch is the backend for Google Stackdriver

Monitors cloud customers and Google services used by those customers

A good deal of important development to do this

- Encryption at rest
- Carefully controlled and audited access
- Different ways of naming things and data model

Background
Architecture and Data Model
Queries
Using Monarch
Monarch Platform
Lessons Learned re: Scaling

Lessons Learned re: Scaling

Maintain Good Hygiene
Scale horizontally -- **only** -- **and it's hard!**
Reduce dimensions early

Lessons Learned - Good Hygiene

Concurrency: don’t make long tails longer.

Periodically assess all components.

Always be deprecating.

Study outliers carefully!

Lessons Learned - Scaling Horizontally

It’s hard, but it’s the only way.

Increase the number of leaves and zones.

Watch out for:

- Centralized services that become bottlenecks.
- Non-constant per-backend costs.
- Query fan-out.

Lessons Learned - Reduce Dimensions Early

Aggregate data as it arrives.

Configuration and data multiplexing are important.

Users must be able to see “through” the aggregation.

Lessons Learned - See through aggregation

Home Query playground No consoles available Settings adammck Change role Help

Hide Editor Query Options Display Refresh Short URL Save ...

/test/adammck/my_web_service/query_latency: region zone

Insert Step

Type: DISTRIBUTION

Window: Align

Operator: NEXT_OLDER

Max interpolation gap: 20m

Explicit output period: 30s

Insert Step

Type: DISTRIBUTION

Group By

Unaligned

Reducer: SUM

Sample as exemplars

Threshold: 0.5

Include dropped fields in exemplars.

Group by:

monarch.BorgTask: borg_job borg_task_num

/test/adammck/my_web_service/query_latency: region zone

Insert Step

Type: DISTRIBUTION

Window: Align

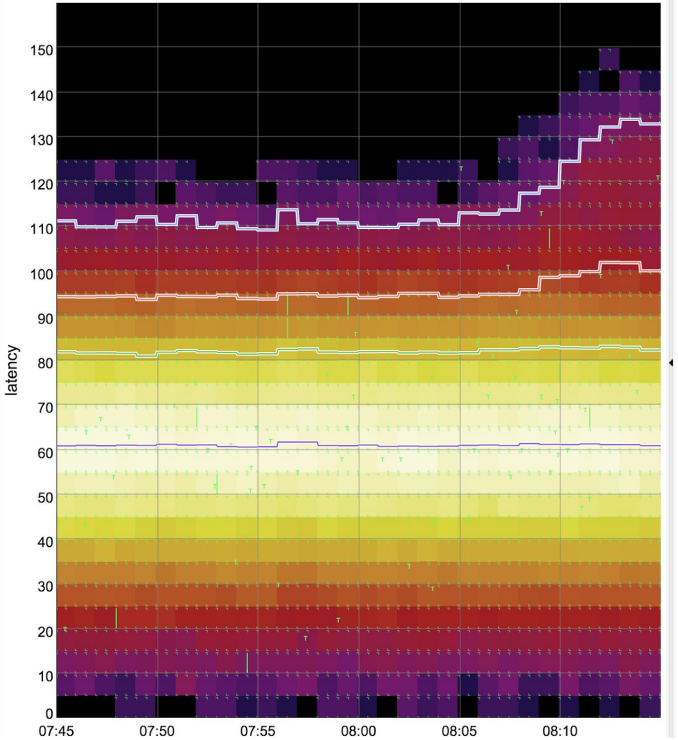
Operator: NEXT_OLDER

Max interpolation gap: 20m

Explicit output period: auto-determined

Insert Step

Type: DISTRIBUTION



Lessons Learned - See through aggregation

Home Query playground No consoles available Settings adammck Change role Help

Hide Editor Query Options Display Refresh Short URL Save ...

/test/adammck/my_web_service/query_latency: region zone

Insert Step

Type: DISTRIBUTION

Window: Align

Operator: NEXT_OLDER

Max interpolation gap: 20m

Explicit output period: 30s

Insert Step

Type: DISTRIBUTION

Group By

Unaligned

Reducer: SUM

Sample as exemplars

Threshold: 0.5

Include dropped fields in exemplars.

Group by:

monarch.BorgTask: borg_job borg_task_num

/test/adammck/my_web_service/query_latency: region zone

Insert Step

Type: DISTRIBUTION

Window: Align

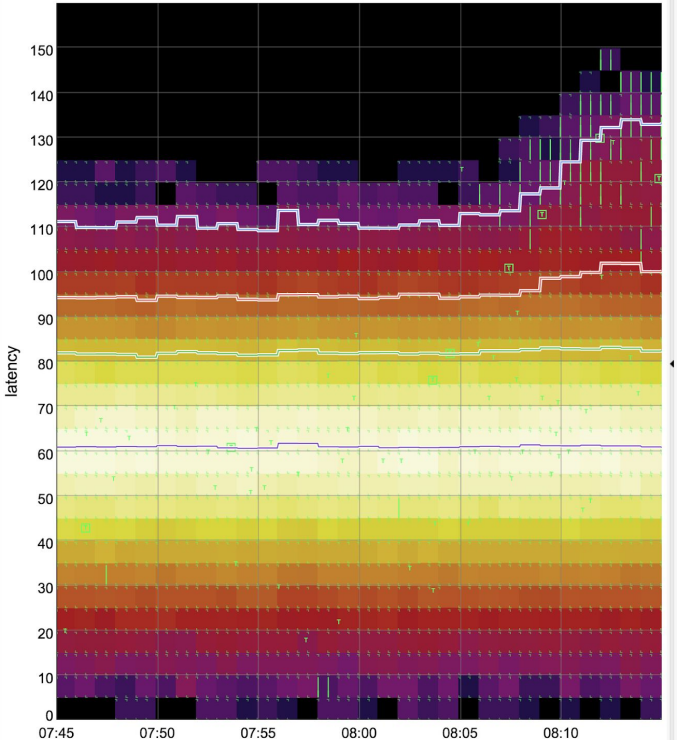
Operator: NEXT_OLDER

Max interpolation gap: 20m

Explicit output period: auto-determined

Insert Step

Type: DISTRIBUTION



Google

Lessons Learned - See through aggregation

Home Query playground No consoles available Settings adammck Change role Help

Hide Editor Query Options Display Refresh Short URL Save ...

/test/adammck/my_web_service/query_latency: region zone

Insert Step

Type: DISTRIBUTION

Window: Align

Operator: NEXT_OLDER

Max interpolation gap: 20m

Explicit output period: 30s

Insert Step

Type: DISTRIBUTION

Group By

Unaligned

Reducer: SUM

Sample as exemplars

Threshold: 0.5

Include dropped fields in exemplars.

Group by:

monarch.BorgTask: borg_job borg_task_num

/test/adammck/my_web_service/query_latency: region zone

Insert Step

Type: DISTRIBUTION

Window: Align

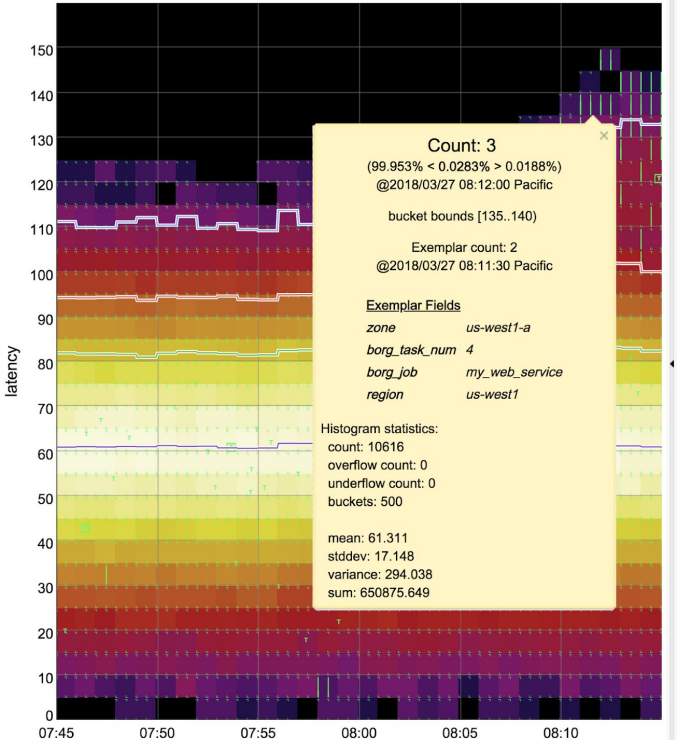
Operator: NEXT_OLDER

Max interpolation gap: 20m

Explicit output period: auto-determined

Insert Step

Type: DISTRIBUTION



Google

Lessons Learned re: Scaling

Maintain Good Hygiene
Scale horizontally -- **only** -- **and it's hard!**
Reduce dimensions early

This is a sampling of lessons we've learned--there are many more.

Thank You