Debugging Microservices:
How Google SREs Resolve Outages

Liz Fong-Jones and Adam Mckaig, Google
@lizthegrey // @adam7mck

#QConNYC 2018
June 28, 2018

Why Production Matters

40%[1] to **90%**[2] of the total costs of software are incurred after launch.

[1] Glass, R. (2002). Facts and Fallacies of Software Engineering, Addison-Wesley Professional; p. 115.
[2] Dehaghani, S. M. H., & Hajrahimi, N. (2013). Which Factors Affect Software Projects Maintenance Cost More? Acta Informatica Medica, 21(1), 63–66. http://doi.org/10.5455/AIM.2012.21.63-66

Why are we here?

- Liz
  - Past: Play Books, Flights, Bigtable, & many other teams
  - Now: Customer Reliability Engineering at Google
- Adam
  - Past: New York Times, Bloomberg, UNICEF
  - Now: Monitoring SRE at Google

Google Cloud

So you have a service...

Let's assume that it exports lots and lots of metrics.

- Microservices!
- Dozens or more hosts
- High cardinality: metrics * hosts > 1000
- Distributed across multiple regions/zones
- Serving latency-sensitive user-initiated queries

... and we're using SLOs...

Why?

- Prioritize user happiness
- Higher signal to noise ratio

How?

- Measure if service is "good enough" (SLIs)
- Set overall top-line targets (SLOs)
- Track performance against targets.

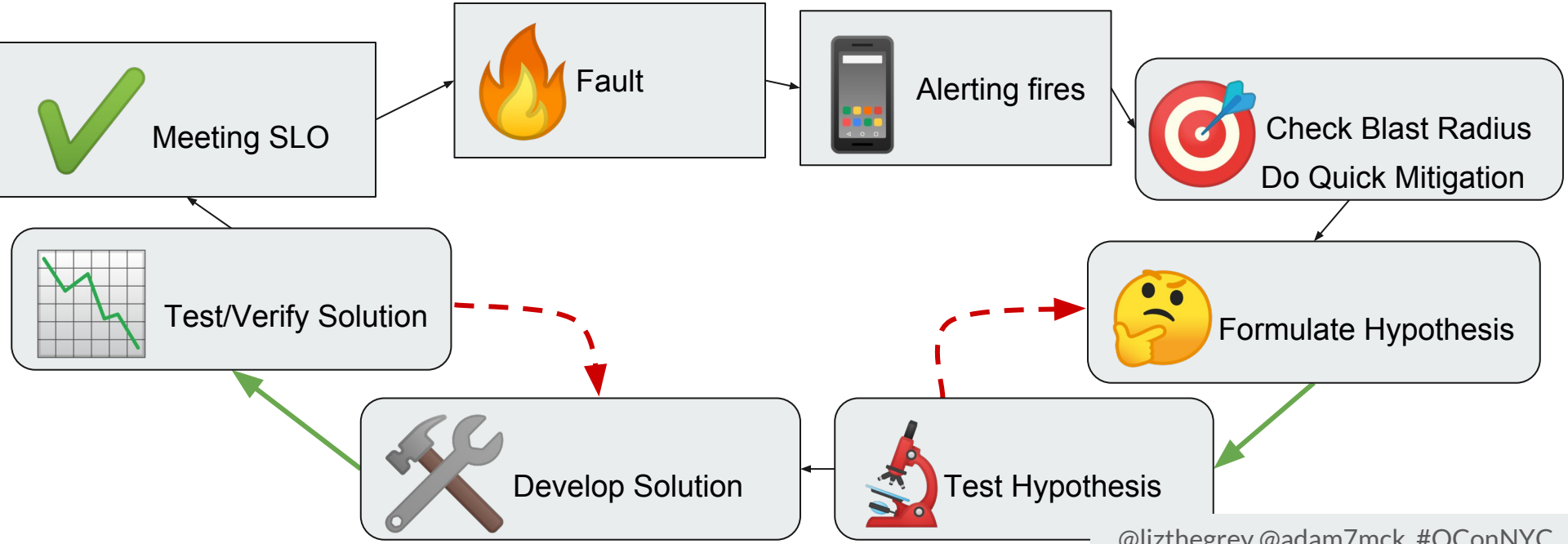
(more on SLOs at https://www.youtube.com/watch?v=tEylFyxbDLE)

... and something goes boom!

- Users see errors
- Error and/or Latency Service Level Indicators start failing
- Error budget starts being burnt
- Service Level Objective is in danger. Pager goes off.
- Now what?

- **Prioritize speed of mitigation and recovery**

The debugging feedback loop:



How can we improve resiliency?

Formulate hypotheses faster and test them faster (core OODA loop)

Let's work an example, using 3 of our best practices:

1. **Layer Peeling:** precomputation and interactive drilldowns
2. **Dynamic Data Joins:** use fewer pre-specified tags
3. **Exemplars:** dtracing + distributions

A note about technologies

- Panopticon/Monarch (Google-internal)
- Techniques implementable anywhere
- We'll name public implementations
- Also coming to Stackdriver!

Error: Percentage of slow user queries too high

```
Env: Production
Region: us-west1

Alert: TooManySlowUserQueries
Since: 2018-01-18 13:30:00 (4s)
URL: http://pcon/KW6UZ4CX

Playbook URL: http://go/my_service_playbook
Silence URL: http://silence/KW6UZ4CX
```

Error: Percentage of slow user queries too high

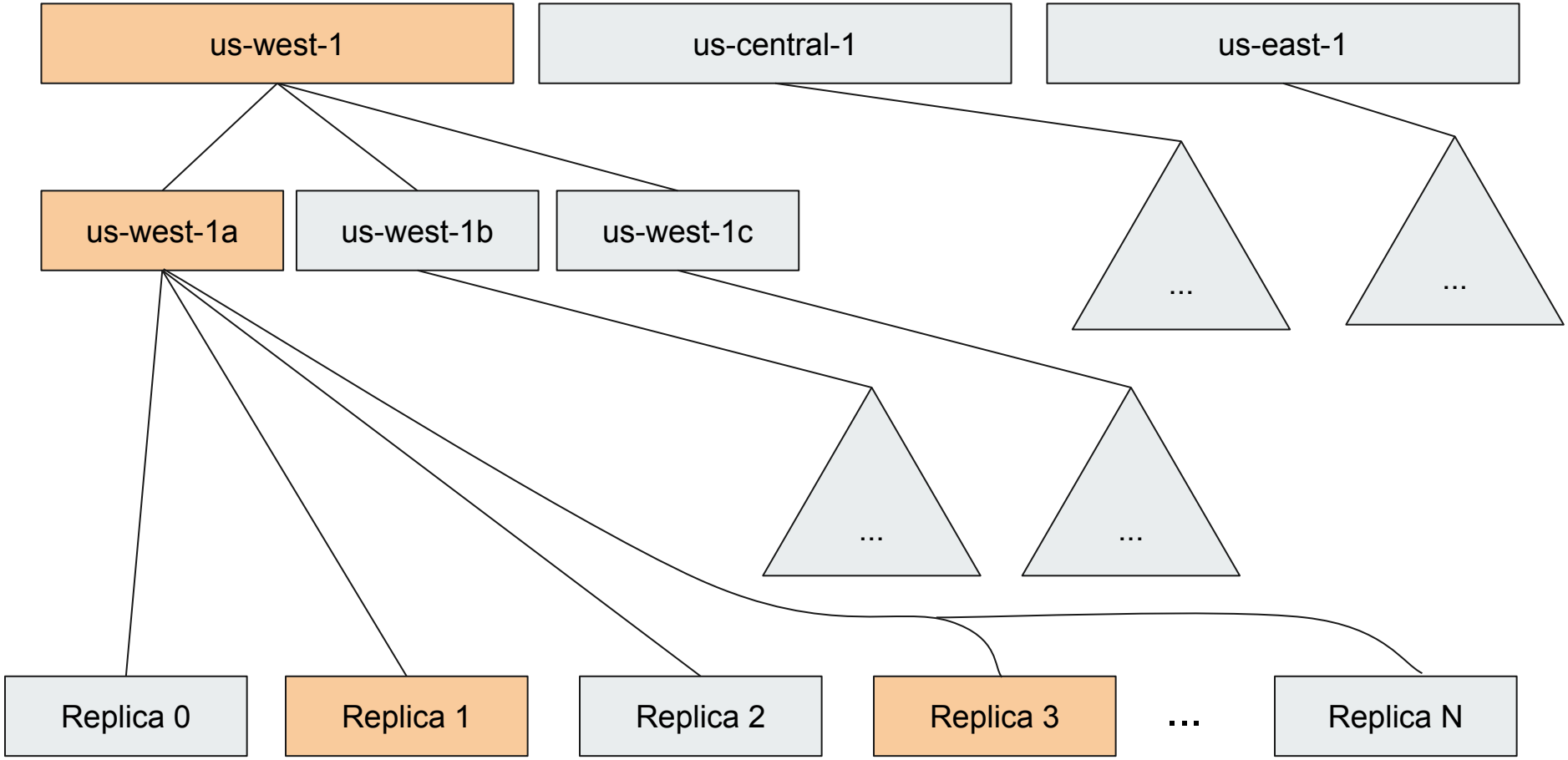
- What can we discover about these slow queries?
- Maybe they’re all being served by the same subset of replicas.
- ...or maybe they’re distributed evenly across all replicas.

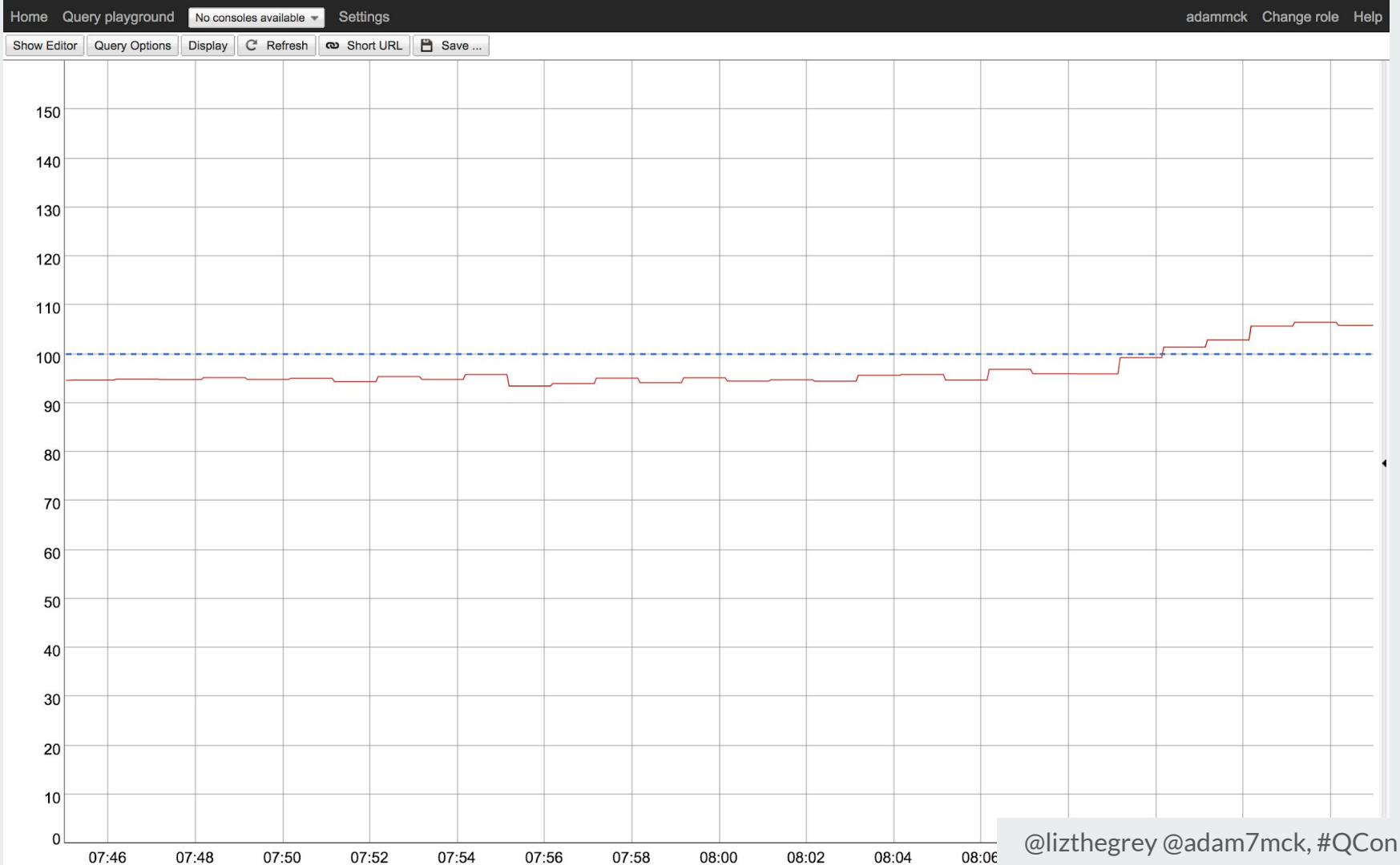
1. Layer Peeling

1. Layer Peeling

Find the problem by peeling the onion back.

1. Start with single (regional) stream for SLI [and check blast radius]
2. Examine latency by zone
3. Filter to affected zones
4. Examine latency by replica
5. Filter to affected replicas
6. Isolate slow replicas


us-west-1
us-central-1
us-east-1
us-west-1a
us-west-1b
us-west-1c
...
...
...
...
Replica 0
Replica 1
Replica 2
Replica 3
...
Replica N


Home
Query playground
No consoles available
Settings
adammck
Change role
Help
Show Editor
Query Options
Display
Refresh
Short URL
Save ...
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
07:46
07:48
07:50
07:52
07:54
07:56
07:58
08:00
08:02
08:04
08:06

Home Query playground No consoles available Settings adammck Change role Help

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

Query 1 + Add Query

Y-axis: Left Right Visible Smoothing: 1 Duplicate

Compare to historical: None Convert units: None

Constant values to show:

- Label: SLO Value: 100

+

Expression Mash

Analyze query execution

Fetch

Align 20m

Group MEAN m:r

Fetch

Source: Precomputed('adammck', '__panopticon', 'mws_latency_98th') precomputed

Filters:

- metric:region == us-west1

+

Replace this with its definition

Role adammck retains this by: 1d@1m

Custom Query

Insert Step

Type: DOUBLE

Window: Align

Operator: NEXT_OLDER

Max interpolation gap: 20m

Explicit output period: auto-determined

Insert Step

Type: DOUBLE

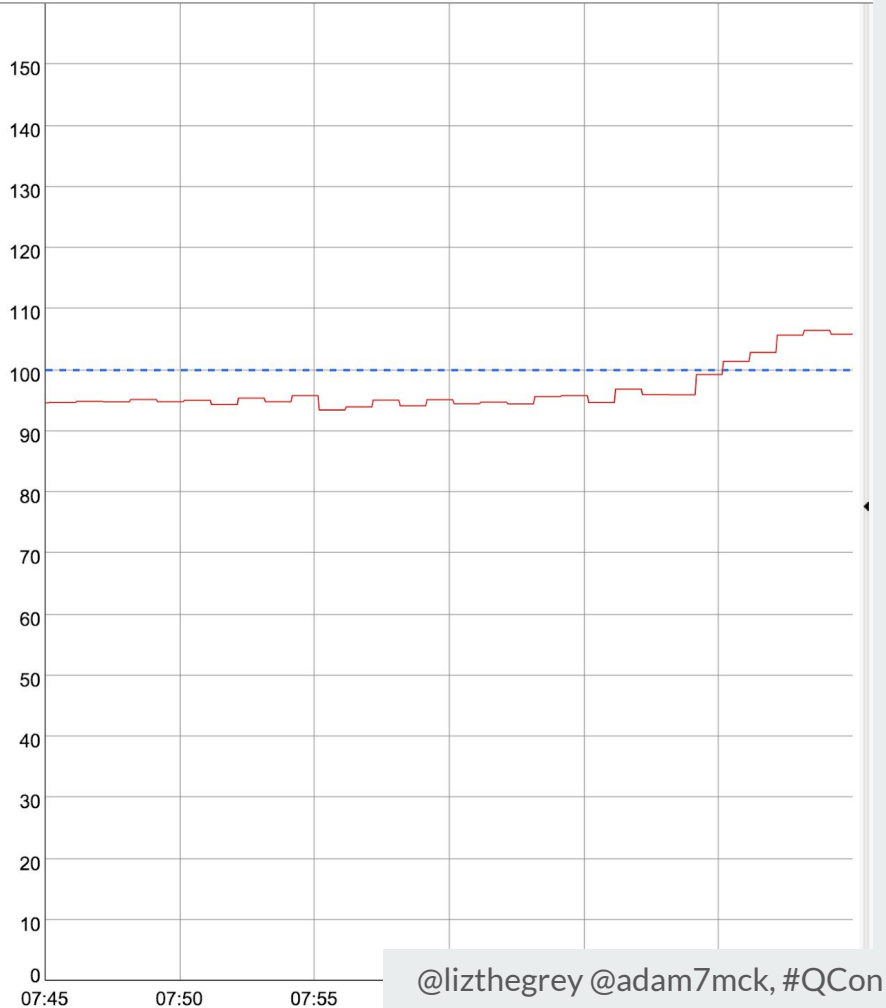
Group By

Unaligned

Reducer: MEAN

Group by:

monarch.BorgTask: borg_job



Fetch
Source: Precomputed('adammck', '__panopticon', 'mws_latency_98th')
precomputed
Filters:
-
metric:region
==
us-west1
+
Replace this with its definition
Role adammck retains this by: 1d@1m

Fetch
Source: Precomputed('adammck', '__panopticon', 'mws_latency_98th')
precomputed
Filters:
+
No filters
Replace this with its definition
Role adammck retains this by: 1d@1m

Home Query playground No consoles available Settings adammck Change role Help

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

Query 1 + Add Query

Y-axis: Left Right Visible Smoothing: 1 Duplicate

Compare to historical: None Convert units: None

Constant values to show:

- Label: SLO Value: 100

+

Expression Mash

Analyze query execution

Fetch

Align 20m

Group MEAN m:r

Fetch

Source: Precomputed('adammck', '__panopticon', 'mws_latency_98th') precomputed

Filters:

+ *No filters*

Replace this with its definition

Role adammck retains this by: 1d@1m

Custom Query

Insert Step

Type: DOUBLE

Window: Align

Operator: NEXT_OLDER

Max interpolation gap: 20m

Explicit output period: auto-determined

Insert Step

Type: DOUBLE

Group By

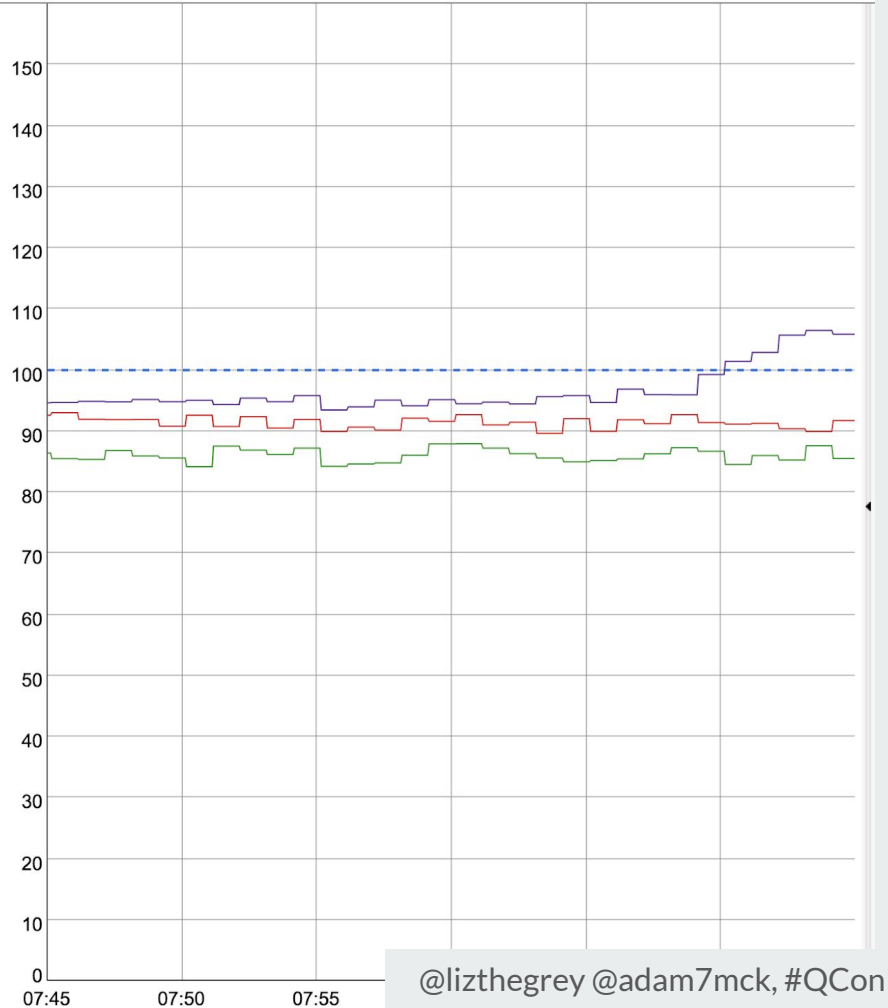
Unaligned

Reducer: MEAN

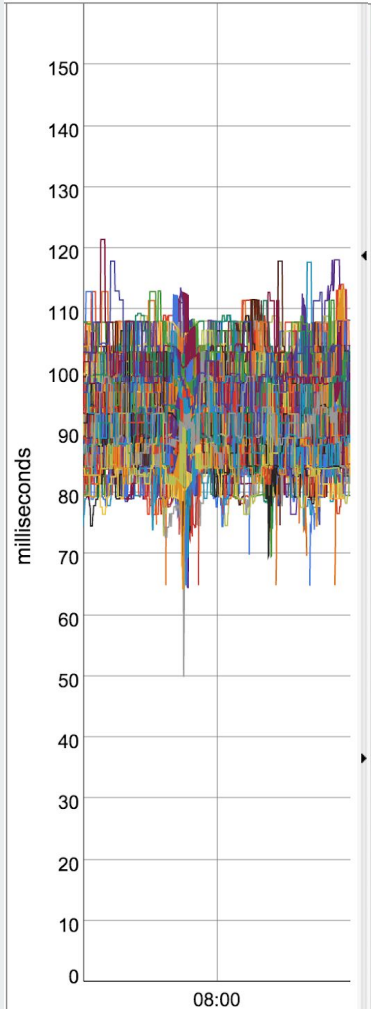
Group by:

monarch.BorgTask: borg_job

Precomputed('adammck', '__panopticon', region zone



Why not look at all the data streams?



Query 1

borg_task_num	region	zone	Latest
0	us-central1	us-central1-a	98.200
0	us-east1	us-east1-a	88.200
0	us-west1	us-west1-a	84.550
0	us-west1	us-west1-b	91.600
0	us-west1	us-west1-c	101.600
1	us-central1	us-central1-a	93.200
1	us-east1	us-east1-a	83.640
1	us-west1	us-west1-a	91.550
1	us-west1	us-west1-b	86.550
1	us-west1	us-west1-c	96.600
2	us-central1	us-central1-a	93.200
2	us-east1	us-east1-a	83.100
2	us-west1	us-west1-a	92.733
2	us-west1	us-west1-b	98.200
2	us-west1	us-west1-c	98.100
3	us-central1	us-central1-a	87.700
3	us-east1	us-east1-a	96.600
3	us-west1	us-west1-a	94.367
3	us-west1	us-west1-b	91.550
3	us-west1	us-west1-c	98.100
4	us-central1	us-central1-a	93.200
4	us-east1	us-east1-a	81.600
4	us-west1	us-west1-a	141.550
4	us-west1	us-west1-b	103.200
4	us-west1	us-west1-c	104.100
5	us-central1	us-central1-a	93.200
5	us-east1	us-east1-a	83.200
5	us-west1	us-west1-a	139.100
5	us-west1	us-west1-b	88.200
5	us-west1	us-west1-c	102.733
6	us-central1	us-central1-a	88.300
6	us-east1	us-east1-a	84
6	us-west1	us-west1-a	88

Back to bisecting...

Home Query playground No consoles available Settings adammck Change role Help

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

Query 1 + Add Query

Y-axis: Left Right Visible Smoothing: 1 Duplicate

Compare to historical: None Convert units: None

Constant values to show:

- Label: SLO Value: 100

+

Expression Mash

Analyze query execution

Fetch

Align 20m

Group MEAN m:r

Fetch

Source: Precomputed('adammck', '__panopticon', 'mws_latency_98th') precomputed

Filters:

- metric:region == us-west1

+

Replace this with its definition

Role adammck retains this by: 1d@1m

Custom Query

Insert Step

Type: DOUBLE

Window: Align

Operator: NEXT_OLDER

Max interpolation gap: 20m

Explicit output period: auto-determined

Insert Step

Type: DOUBLE

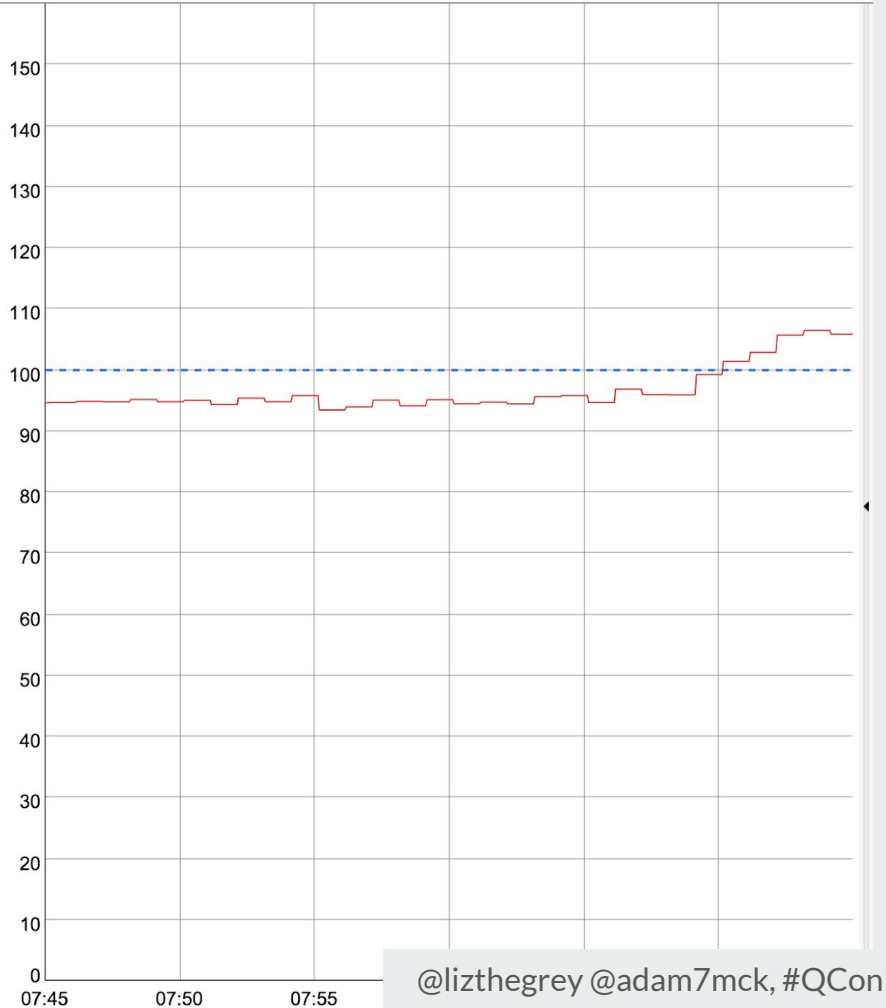
Group By

Unaligned

Reducer: MEAN

Group by:

monarch.BorgTask: borg_job



Home Query playground No consoles available Settings adammck Change role Help

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

Query 1 + Add Query

Y-axis: Left Right Visible Smoothing: 1 Duplicate

Compare to historical: None Convert units: None

Constant values to show:

- Label: SLO Value: 100

+

Expression Mash

Analyze query execution

Fetch

Align 20m

Group MEAN m:r

Fetch

Source: Precomputed('adammck', '__panopticon', 'mws_latency_98th') precomputed

Filters:

- metric:region == us-west1

+

Replace this with its definition

Role adammck retains this by: 1d@1m

Custom Query

Insert Step

Type: DOUBLE

Window: Align

Operator: NEXT_OLDER

Max interpolation gap: 20m

Explicit output period: auto-determined

Insert Step

Type: DOUBLE

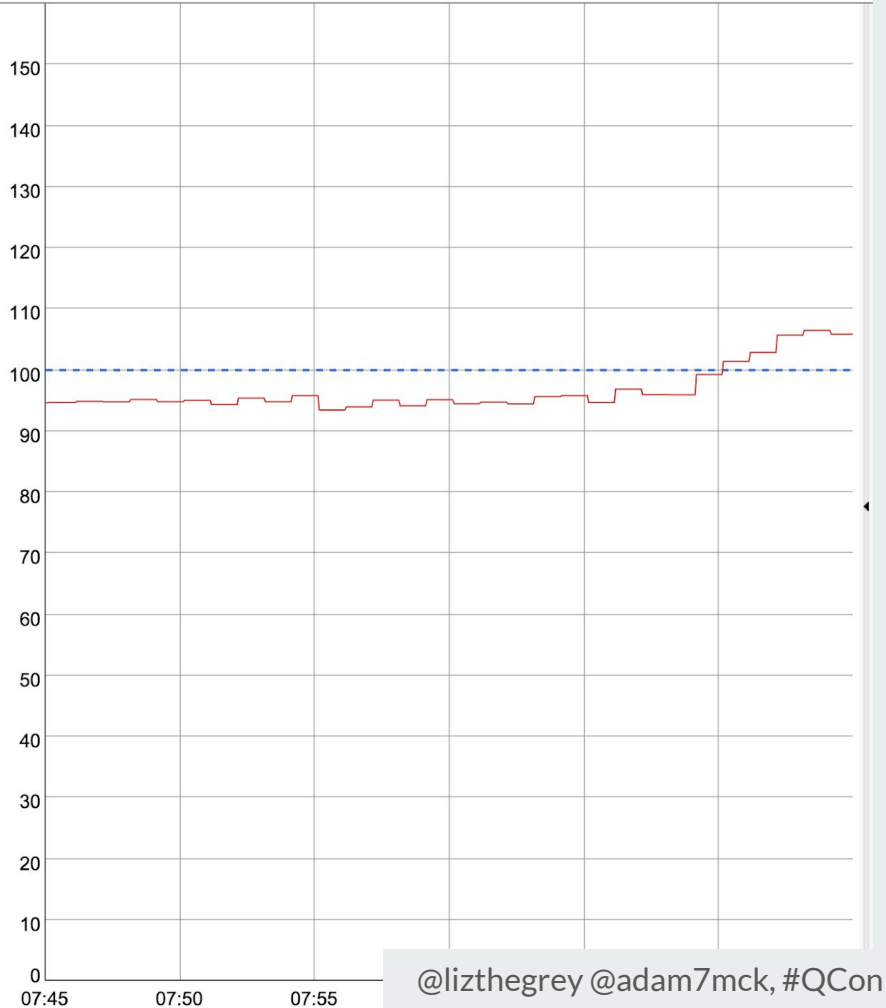
Group By

Unaligned

Reducer: MEAN

Group by:

monarch.BorgTask: borg_job



Home Query playground No consoles available Settings adammck Change role Help

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

Query 1 + Add Query

Y-axis: Left Right Visible Smoothing: 1 Duplicate

Compare to historical: None Convert units: None

Constant values to show:

- Label: SLO Value: 100

+

Expression Mash

Analyze query execution

Fetch

Align 20m

Group SUM b_j, m:r, m:z

Point Percentile(9...

Align 20m

Group MEAN m:r

Fetch

Source: Precomputed('adammck', '__panopticon', 'mws_latency')

precomputed

Filters:

- metric:region == us-west1

+

Replace this with its definition

Role adammck retains this by: 1d@1m

Custom Query

Insert Step

Type: DISTRIBUTION

Window: Align

Operator: NEXT_OLDER

Max interpolation gap: 20m

Explicit output period: 1m

Insert Step

Type: DISTRIBUTION

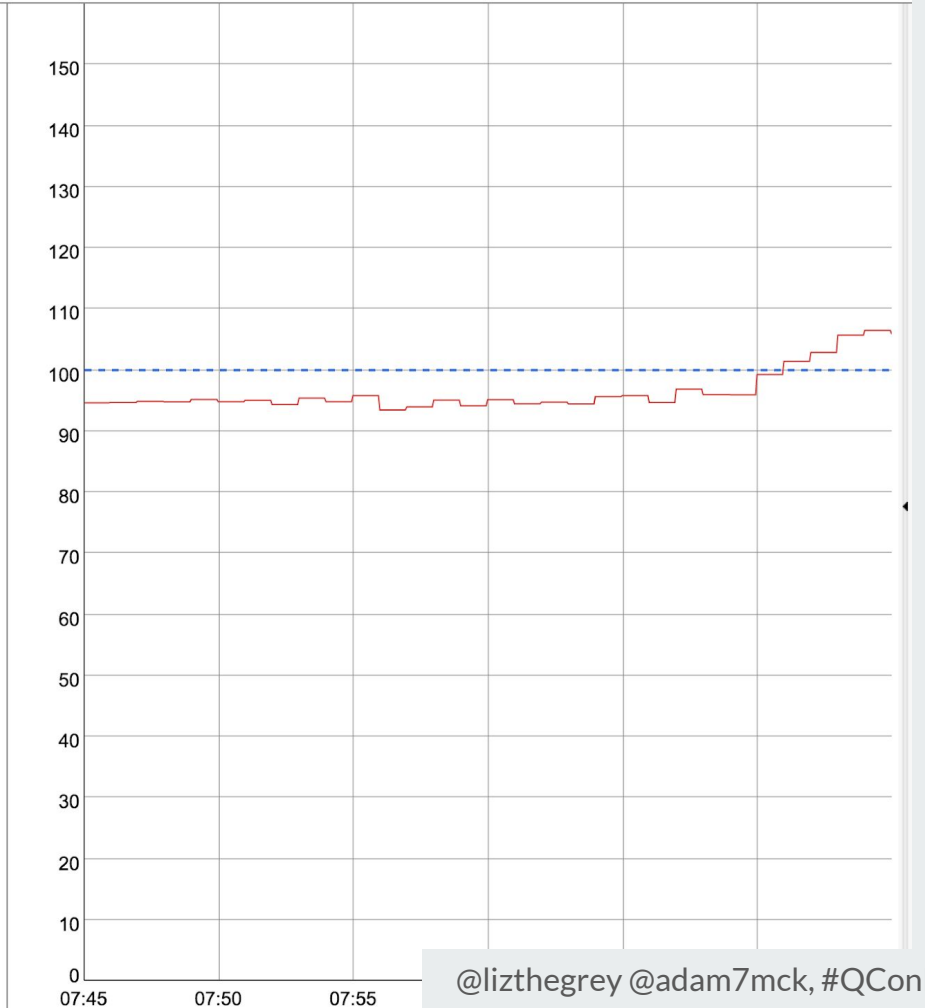
Group By

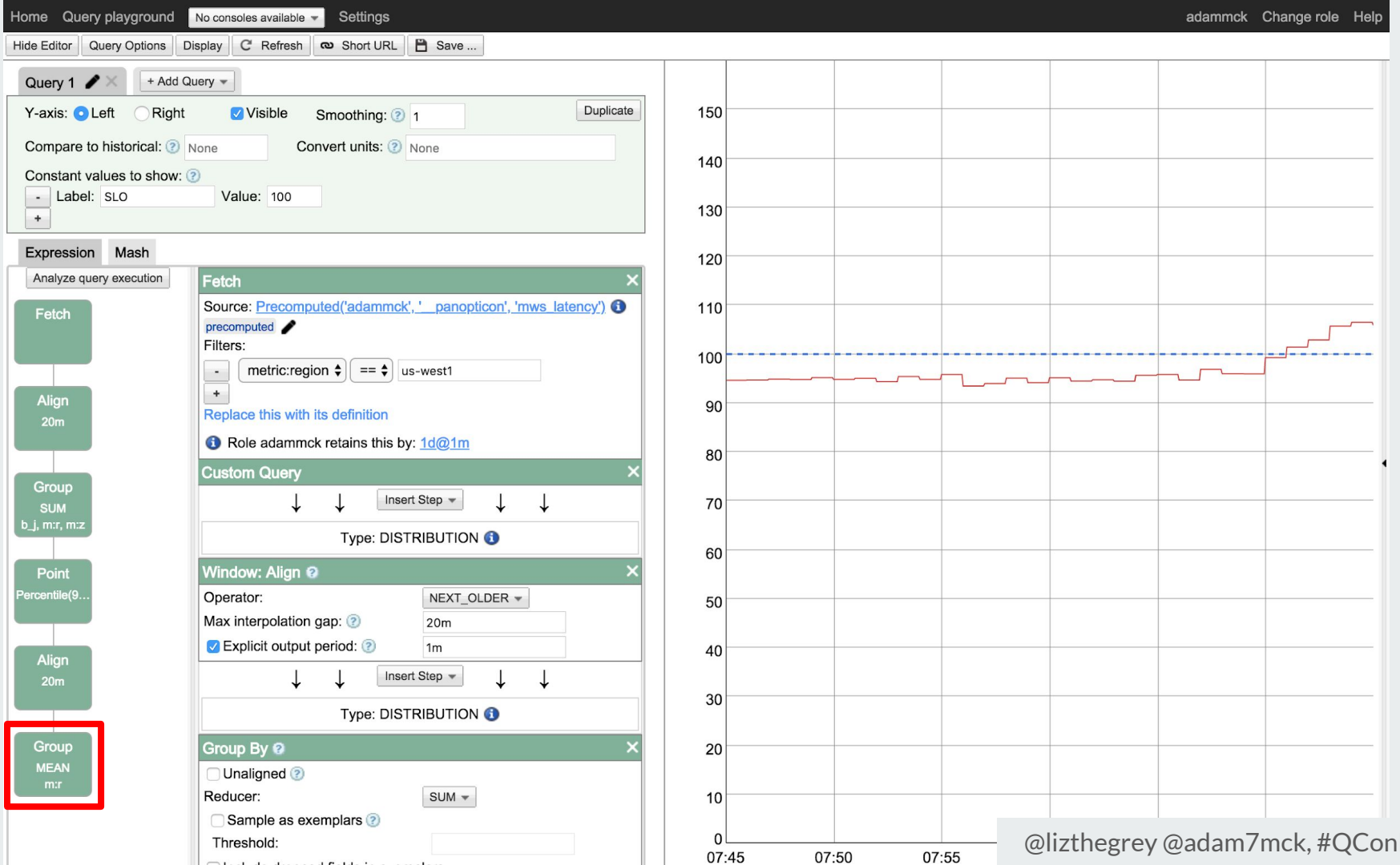
Unaligned

Reducer: SUM

Sample as exemplars

Threshold:




Home
Query playground
No consoles available
Settings
adammck
Change role
Help
Hide Editor
Query Options
Display
Refresh
Short URL
Save ...
Query 1
+ Add Query
Y-axis: Left Right Visible
Smoothing: 1
Duplicate
Compare to historical: None
Convert units: None
Constant values to show:
Label: SLO
Value: 100
Expression
Mash
Analyze query execution
Fetch
Align 20m
Group SUM b_j, m:r, m:z
Point Percentile(9...
Align 20m
Group MEAN m:r
Fetch
Source: Precomputed('adammck', '__panopticon', 'mws_latency')
precomputed
Filters:
metric:region == us-west1
Replace this with its definition
Role adammck retains this by: 1d@1m
Custom Query
Insert Step
Type: DISTRIBUTION
Window: Align
Operator: NEXT_OLDER
Max interpolation gap: 20m
Explicit output period: 1m
Insert Step
Type: DISTRIBUTION
Group By
Unaligned
Reducer: SUM
Sample as exemplars
Threshold:
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
07:45
07:50
07:55

Group By
Unaligned
Reducer:
MEAN
Group by:
monarch.BorgTask:
borg_job
Precomputed('adammck', '__panopticon',
'mws_latency'):
region
zone

Home Query playground No consoles available Settings adammck Change role Help

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

Include dropped fields in exemplars.

Group by:

monarch.BorgTask: borg_job

Precomputed('adammck', '__panopticon', 'mws_latency'): region zone

Insert Step

Type: DISTRIBUTION

Point Operation

Operation: Percentile(98, VAL)

Insert Step

Type: DOUBLE

Window: Align

Operator: NEXT_OLDER

Max interpolation gap: 20m

Explicit output period: auto-determined

Insert Step

Type: DOUBLE

Group By

Unaligned

Reducer: MEAN

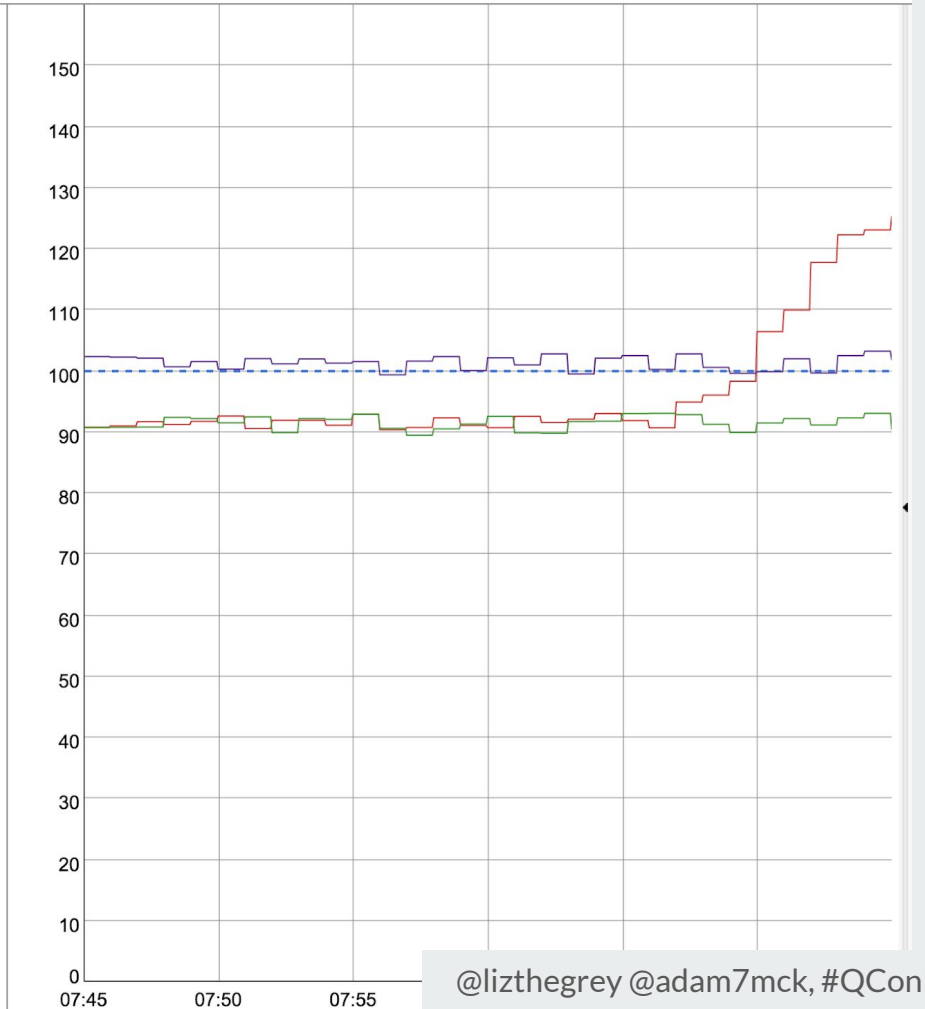
Group by:

monarch.BorgTask: borg_job

Precomputed('adammck', '__panopticon', 'mws_latency'): region zone

Insert Step

Type: DOUBLE



Home Query playground No consoles available Settings adammck Change role Help

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

Include dropped fields in exemplars.

Group by:

monarch.BorgTask: borg_job

Precomputed('adammck', '__panopticon', 'mws_latency'): region zone

Insert Step

Type: DISTRIBUTION

Point Operation

Operation: Percentile(98, VAL)

Insert Step

Type: DOUBLE

Window: Align

Operator: NEXT_OLDER

Max interpolation gap: 20m

Explicit output period: auto-determined

Insert Step

Type: DOUBLE

Group By

Unaligned

Reducer: MEAN

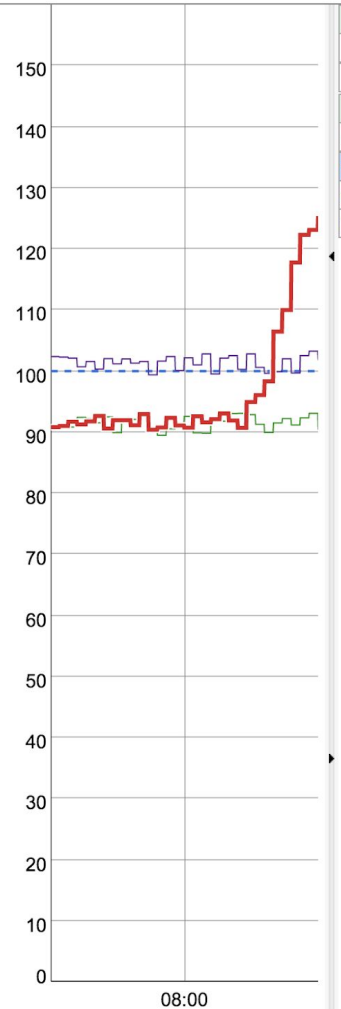
Group by:

monarch.BorgTask: borg_job

Precomputed('adammck', '__panopticon', 'mws_latency'): region zone

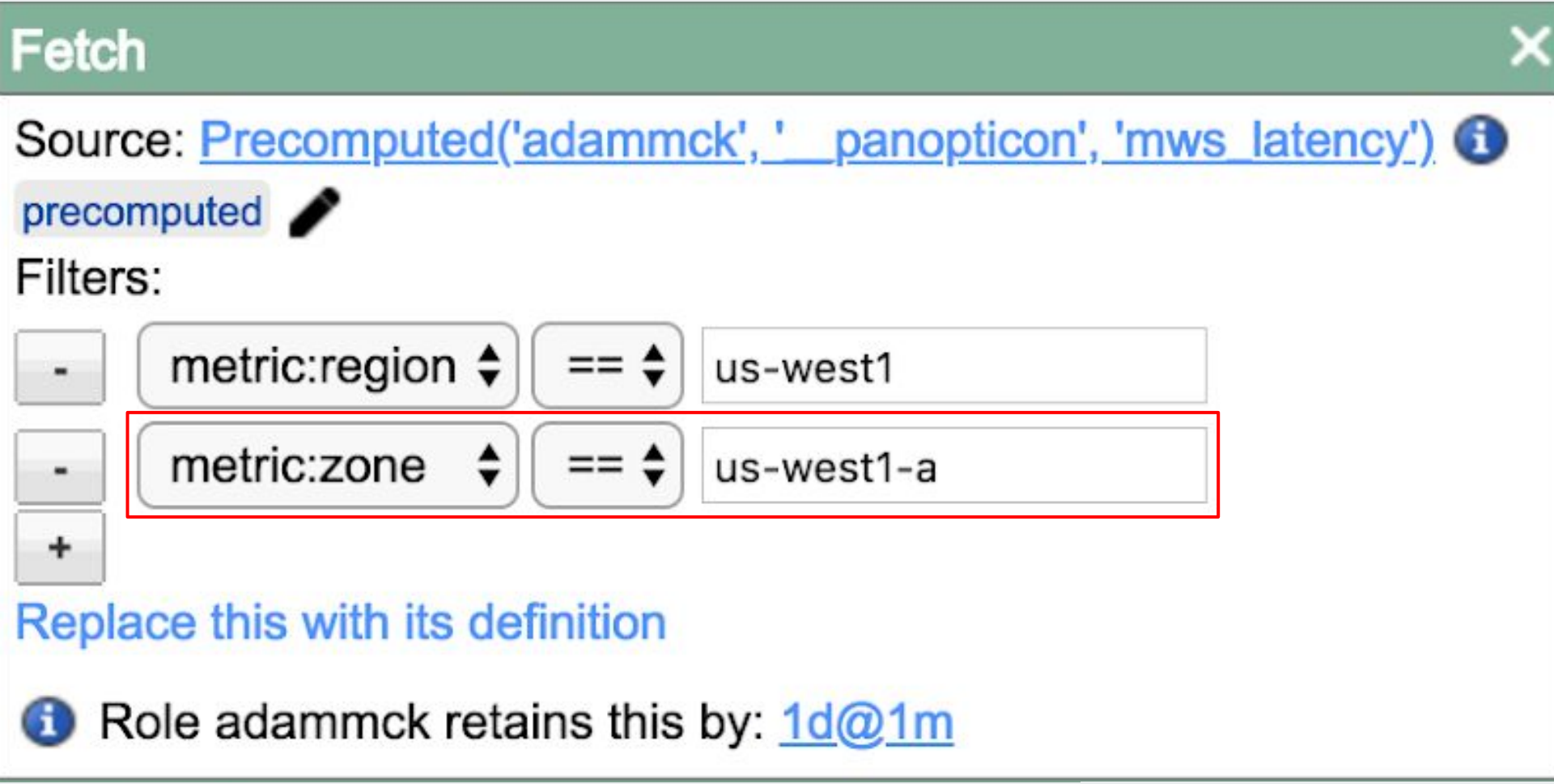
Insert Step

Type: DOUBLE



[constant] SLO
Value
100.000

Query 1		
region	zone	Latest
us-west1	us-west1-a	125.391
us-west1	us-west1-b	90.409
us-west1	us-west1-c	101.812


Fetch
Source: Precomputed('adammck', '__panopticon', 'mws_latency')
precomputed
Filters:
- metric:region == us-west1
- metric:zone == us-west1-a
+
Replace this with its definition
Role adammck retains this by: 1d@1m

Home Query playground No consoles available Settings adammck Change role Help

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

Query 1 + Add Query

Y-axis: Left Right Visible Smoothing: 1 Duplicate

Compare to historical: None Convert units: None

Constant values to show:

- Label: SLO Value: 100

+

Expression Mash

Analyze query execution

Fetch

Align 20m

Group SUM b_j, m:r, m:z

Point Percentile(9...

Align 20m

Group MEAN m:r, m:z

Fetch

Source: Precomputed('adammck', '__panopticon', 'mws_latency')

precomputed

Filters:

- metric:region == us-west1
- metric:zone == us-west1-a

+

Replace this with its definition

Role adammck retains this by: 1d@1m

Custom Query

Insert Step

Type: DISTRIBUTION

Window: Align

Operator: NEXT_OLDER

Max interpolation gap: 20m

Explicit output period: 1m

Insert Step

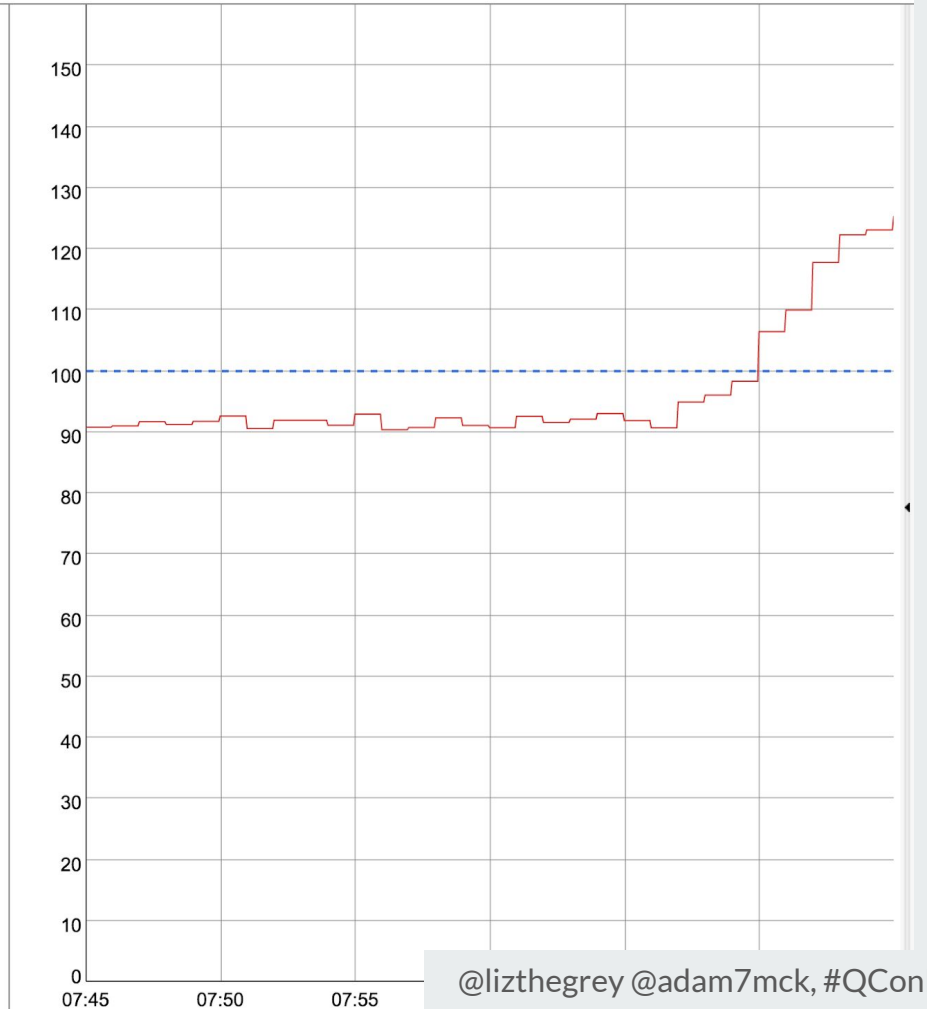
Type: DISTRIBUTION

Group By

Unaligned

Reducer: SUM

Sample as exemplars



Home Query playground No consoles available Settings adammck Change role Help

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

Group
SUM
b_j, m:r, m:z

Point
Percentile(9...

Replace this with its definition

Role adammck retains this by: 1d@1m

Custom Query

Insert Step

Type: DISTRIBUTION

Window: Align

Operator: NEXT_OLDER

Max interpolation gap: 20m

Explicit output period: auto-determined

Insert Step

Type: DISTRIBUTION

Group By

Unaligned

Reducer: SUM

Sample as exemplars

Threshold:

Include dropped fields in exemplars.

Group by:

monarch.BorgTask: borg_job

Precomputed('adammck', '__panopticon', 'mws_latency'): region zone

Insert Step

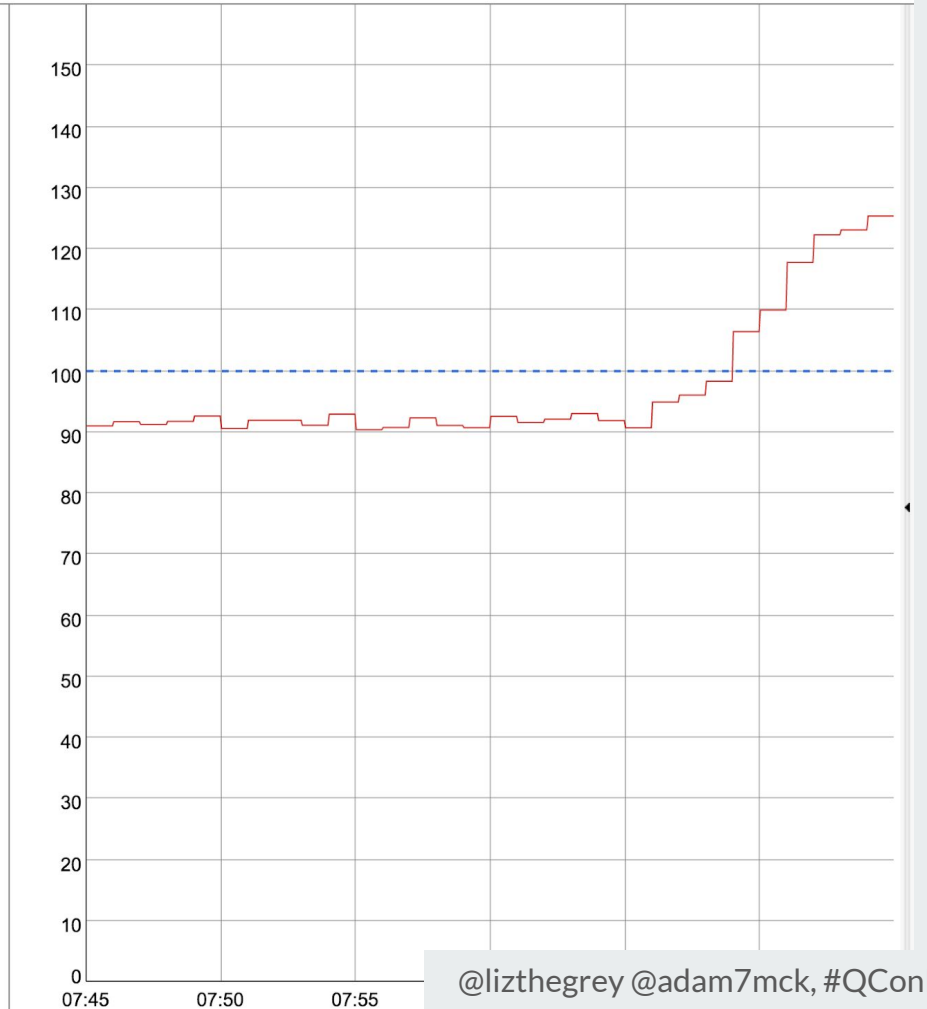
Type: DISTRIBUTION

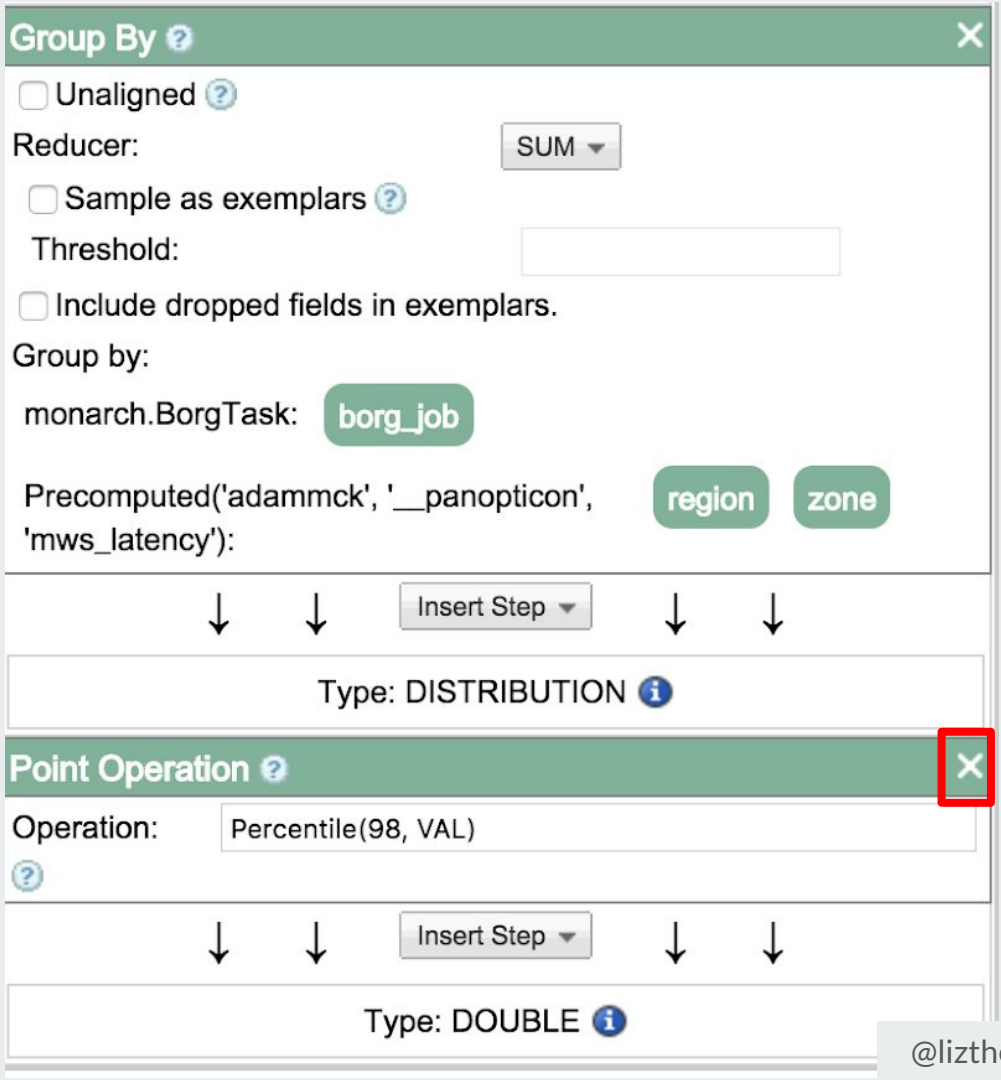
Point Operation

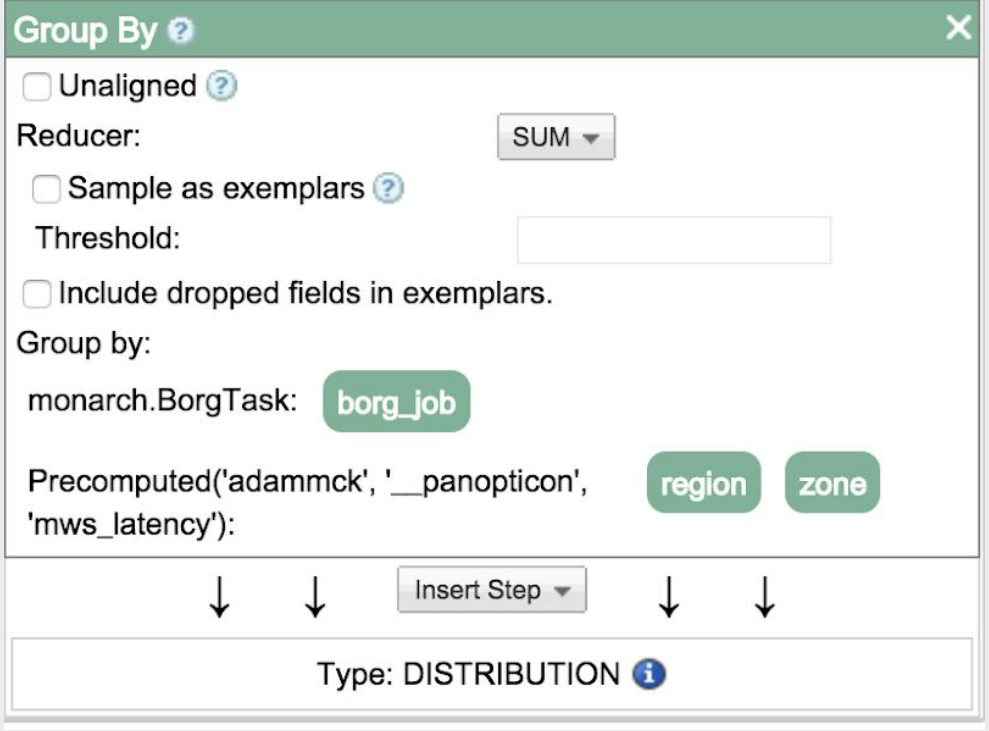
Operation: Percentile(98, VAL)

Insert Step

Type: DOUBLE




Group By
Unaligned
Reducer:
SUM
Sample as exemplars
Threshold:
Include dropped fields in exemplars.
Group by:
monarch.BorgTask:
borg_job
Precomputed('adammck', '__panopticon', 'mws_latency'):
region
zone
Insert Step
Type: DISTRIBUTION
Point Operation
Operation:
Percentile(98, VAL)
Insert Step
Type: DOUBLE


Group By
Unaligned
Reducer:
SUM
Sample as exemplars
Threshold:
Include dropped fields in exemplars.
Group by:
monarch.BorgTask:
borg_job
Precomputed('adammck', '__panopticon', 'mws_latency'):
region
zone
Insert Step
Type: DISTRIBUTION

Home Query playground No consoles available Settings adammck Change role Help

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

Fetch

Align
20m

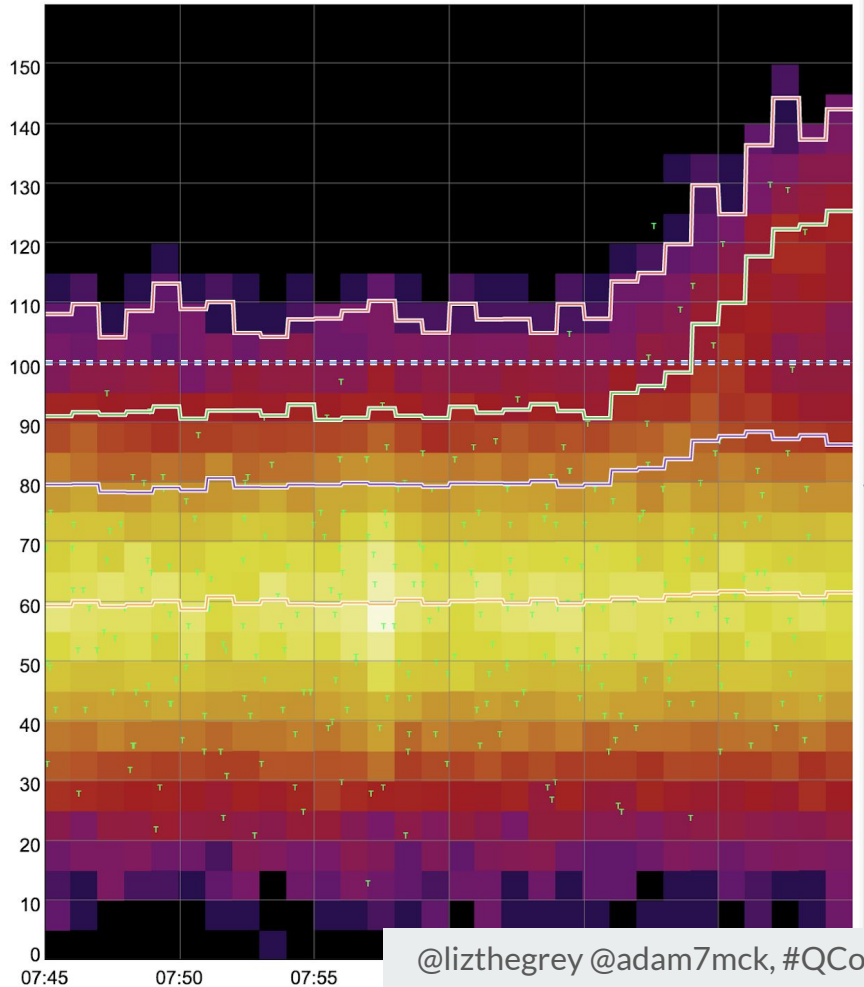
Group
SUM
b_j, m:r, m:z

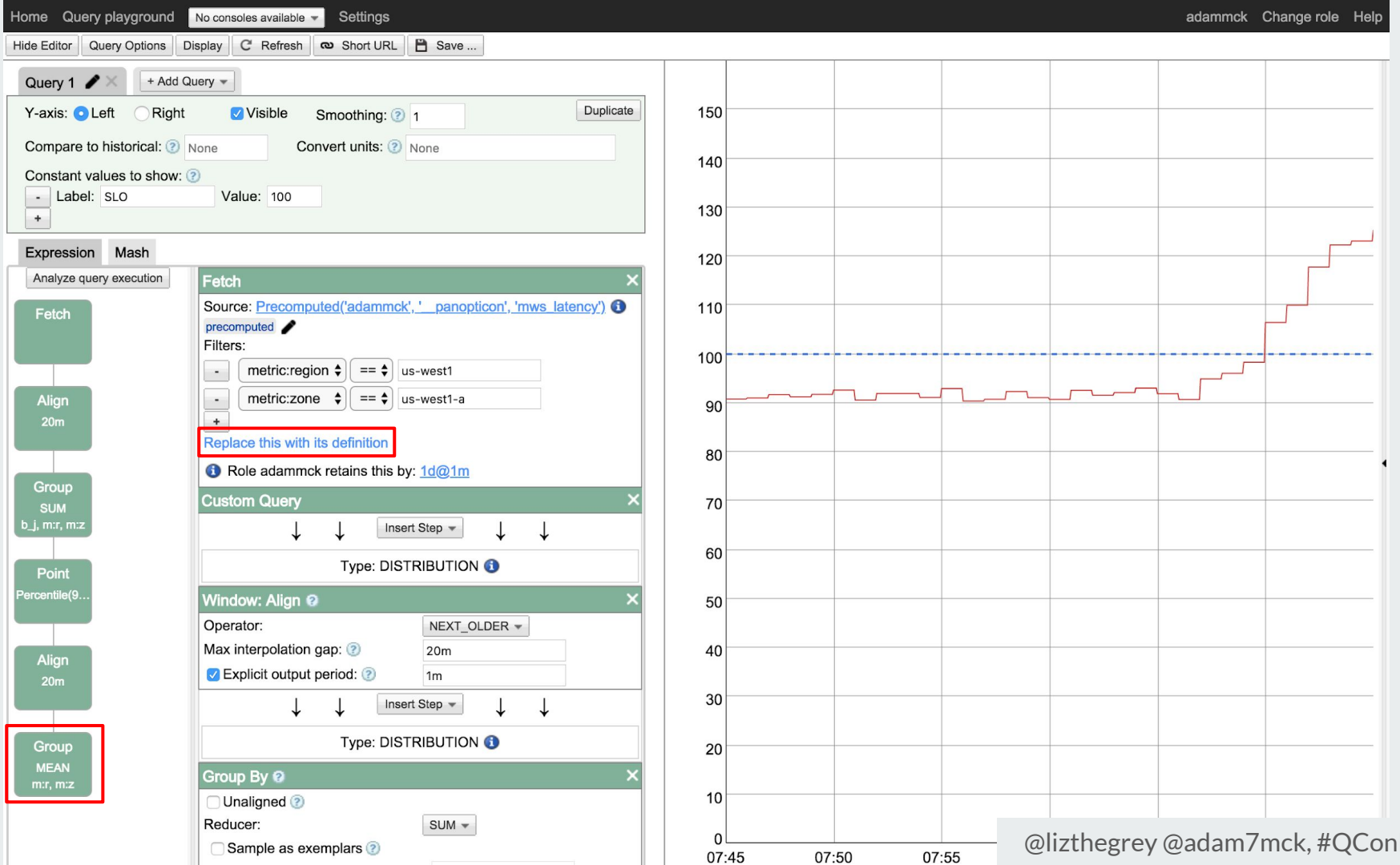
Source: Precomputed('adammck', '__panopticon', 'mws_latency')
precomputed
Filters:
- metric:region == us-west1
- metric:zone == us-west1-a
+
Replace this with its definition
Role adammck retains this by: 1d@1m

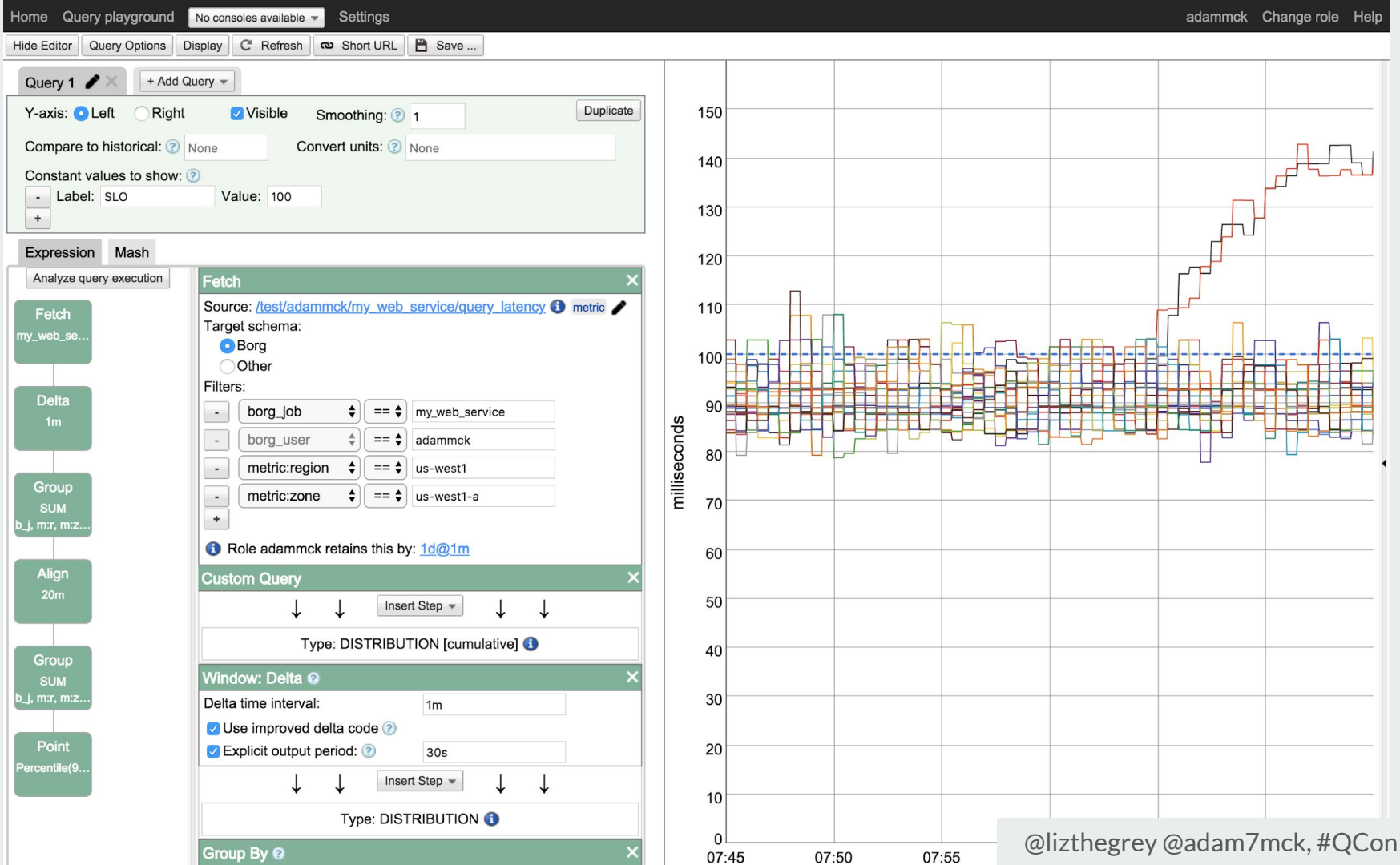
Custom Query
Insert Step
Type: DISTRIBUTION

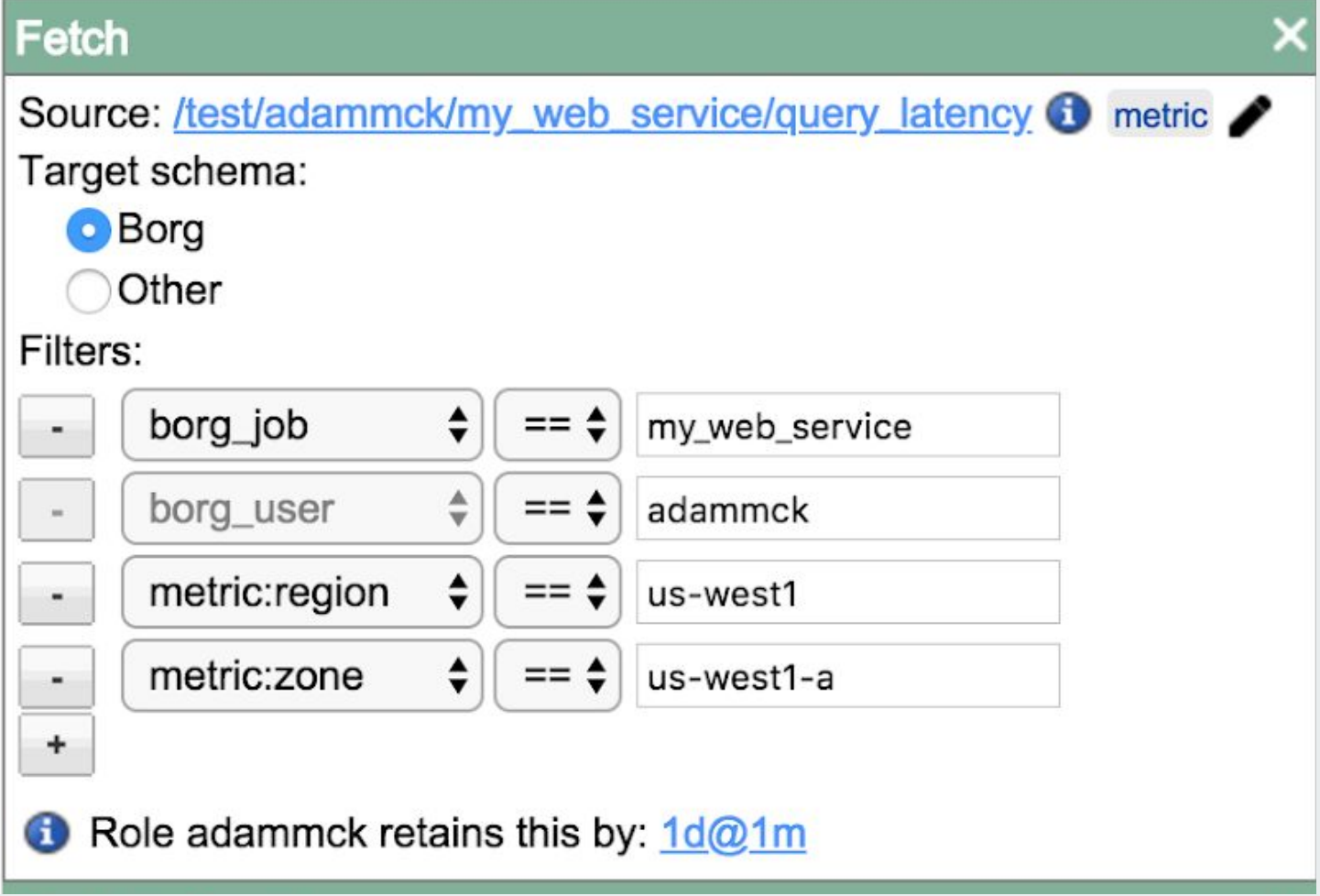
Window: Align
Operator: NEXT_OLDER
Max interpolation gap: 20m
Explicit output period: auto-determined
Insert Step
Type: DISTRIBUTION

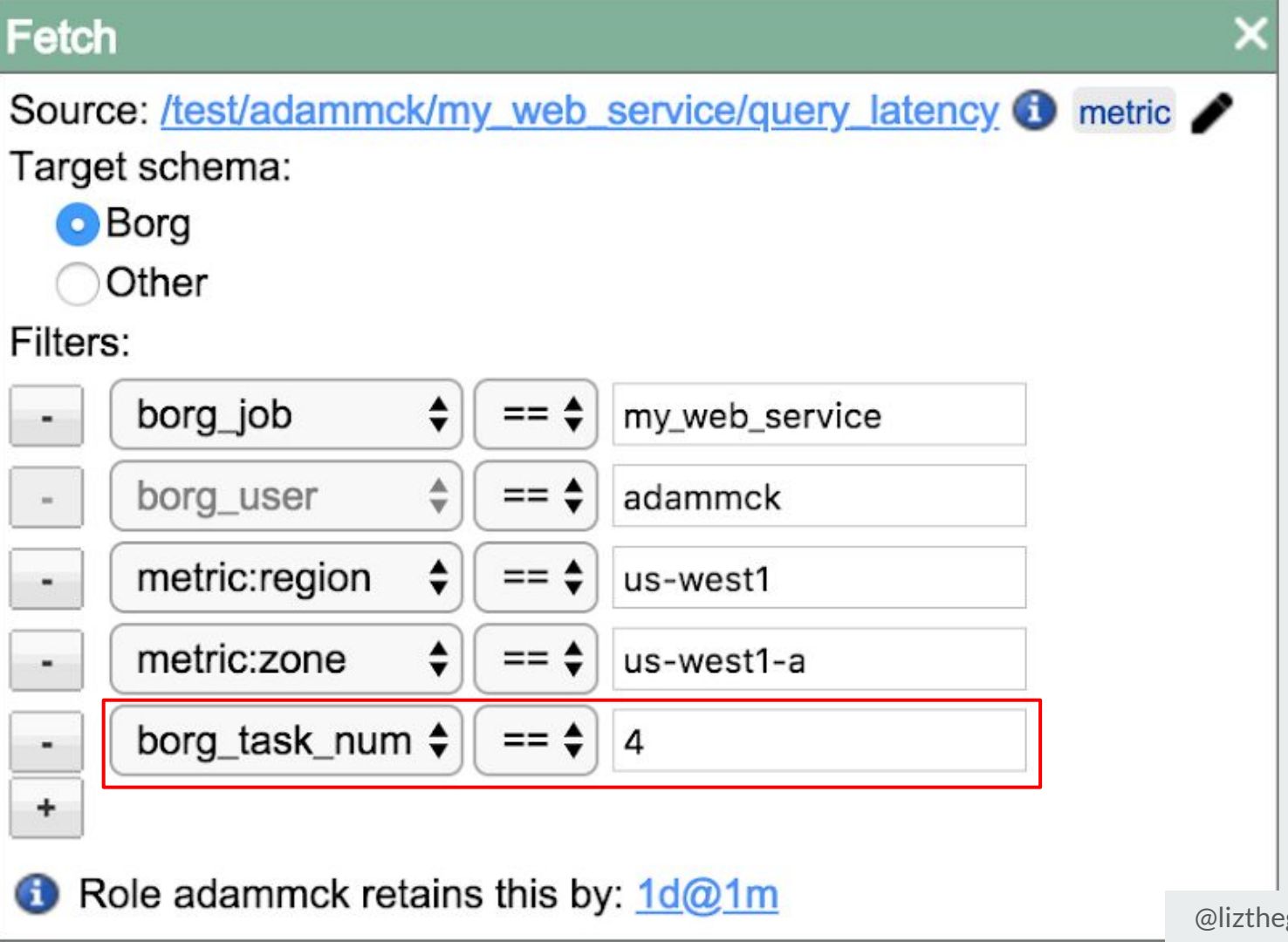
Group By
Unaligned
Reducer: SUM
Sample as exemplars
Threshold:
Include dropped fields in exemplars.
Group by:
monarch.BorgTask: borg_job
Precomputed('adammck', '__panopticon', 'mws_latency'): region zone
Insert Step
Type: DISTRIBUTION




Home
Query playground
No consoles available
Settings
adammck
Change role
Help
Hide Editor
Query Options
Display
Refresh
Short URL
Save ...
Query 1
+ Add Query
Y-axis: Left Right Visible
Smoothing: 1
Duplicate
Compare to historical: None
Convert units: None
Constant values to show:
Label: SLO
Value: 100
Expression
Mash
Analyze query execution
Fetch
Align 20m
Group SUM b_j, m:r, m:z
Point Percentile(9...
Align 20m
Group MEAN m:r, m:z
Fetch
Source: Precomputed('adammck', '__panopticon', 'mws_latency')
precomputed
Filters:
metric:region == us-west1
metric:zone == us-west1-a
Replace this with its definition
Role adammck retains this by: 1d@1m
Custom Query
Insert Step
Type: DISTRIBUTION
Window: Align
Operator: NEXT_OLDER
Max interpolation gap: 20m
Explicit output period: 1m
Insert Step
Type: DISTRIBUTION
Group By
Unaligned
Reducer: SUM
Sample as exemplars
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
07:45
07:50
07:55


Home
Query playground
No consoles available
Settings
adammck
Change role
Help
Hide Editor
Query Options
Display
Refresh
Short URL
Save ...
Query 1
+ Add Query
Y-axis: Left Right Visible
Smoothing: 1
Duplicate
Compare to historical: None
Convert units: None
Constant values to show:
Label: SLO
Value: 100
Expression
Mash
Analyze query execution
Fetch
my_web_se...
Delta
1m
Group
SUM
b_j, m:r, m:z...
Align
20m
Group
SUM
b_j, m:r, m:z...
Point
Percentile(9...
Fetch
Source: /test/adammck/my_web_service/query_latency
metric
Target schema:
Borg
Other
Filters:
borg_job == my_web_service
borg_user == adammck
metric:region == us-west1
metric:zone == us-west1-a
Role adammck retains this by: 1d@1m
Custom Query
Insert Step
Type: DISTRIBUTION [cumulative]
Window: Delta
Delta time interval: 1m
Use improved delta code
Explicit output period: 30s
Insert Step
Type: DISTRIBUTION
Group By
milliseconds
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
07:45
07:50
07:55


Fetch
Source: /test/adammck/my_web_service/query_latency
metric
Target schema:
Borg
Other
Filters:
- borg_job == my_web_service
- borg_user == adammck
- metric:region == us-west1
- metric:zone == us-west1-a
+
Role adammck retains this by: 1d@1m


Fetch
Source: /test/adammck/my_web_service/query_latency
metric
Target schema:
Borg
Other
Filters:
borg_job == my_web_service
borg_user == adammck
metric:region == us-west1
metric:zone == us-west1-a
borg_task_num == 4
Role adammck retains this by: 1d@1m

Home Query playground No consoles available Settings adammck Change role Help

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

Query 1 + Add Query

Y-axis: Left Right Visible Smoothing: 1 Duplicate

Compare to historical: None Convert units: None

Constant values to show:

- Label: SLO Value: 100

+

Expression Mash

Analyze query execution

Fetch my_web_se...

Delta 1m

Group SUM b_j, m:r, m:z...

Align 20m

Group SUM b_j, m:r, m:z...

Point Percentile(9...

Fetch

Source: /test/adammck/my_web_service/query_latency metric

Target schema:

Borg

Other

Filters:

-	borg_job	==	my_web_service
-	borg_user	==	adammck
-	metric:region	==	us-west1
-	metric:zone	==	us-west1-a
-	borg_task_num	==	4

+

Role adammck retains this by: 1d@1m

Custom Query

Insert Step

Type: DISTRIBUTION [cumulative]

Window: Delta

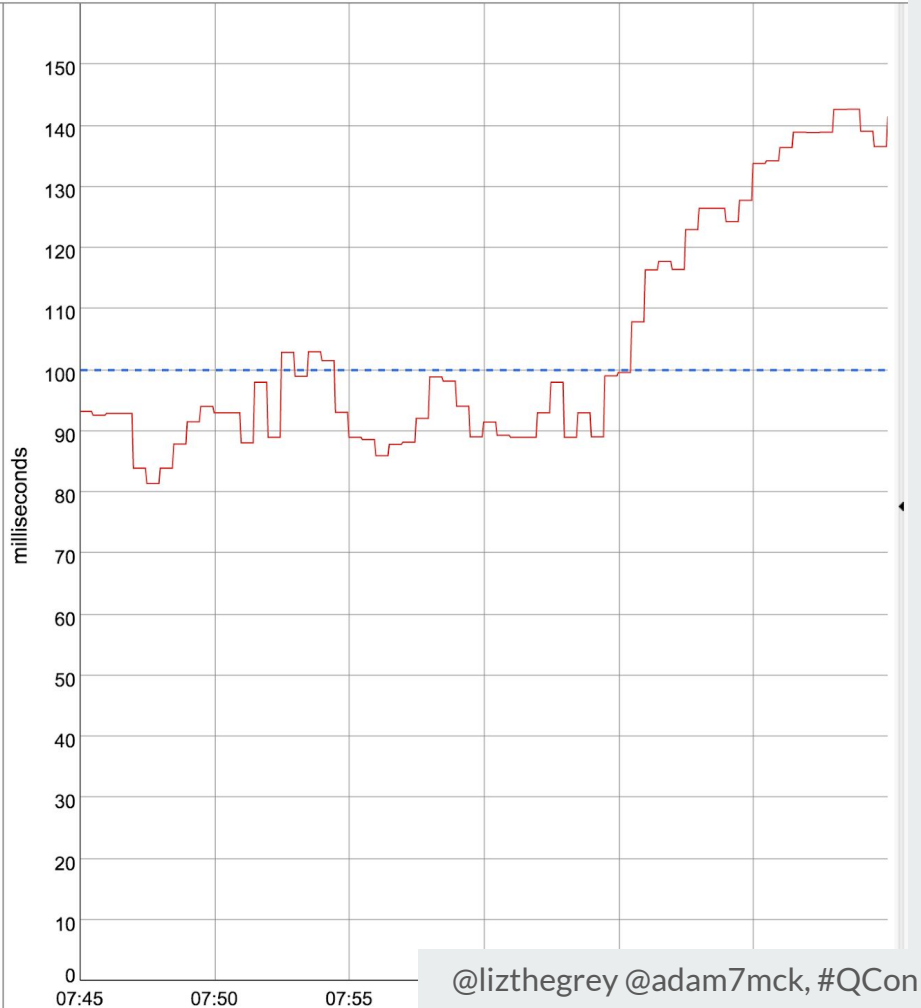
Delta time interval: 1m

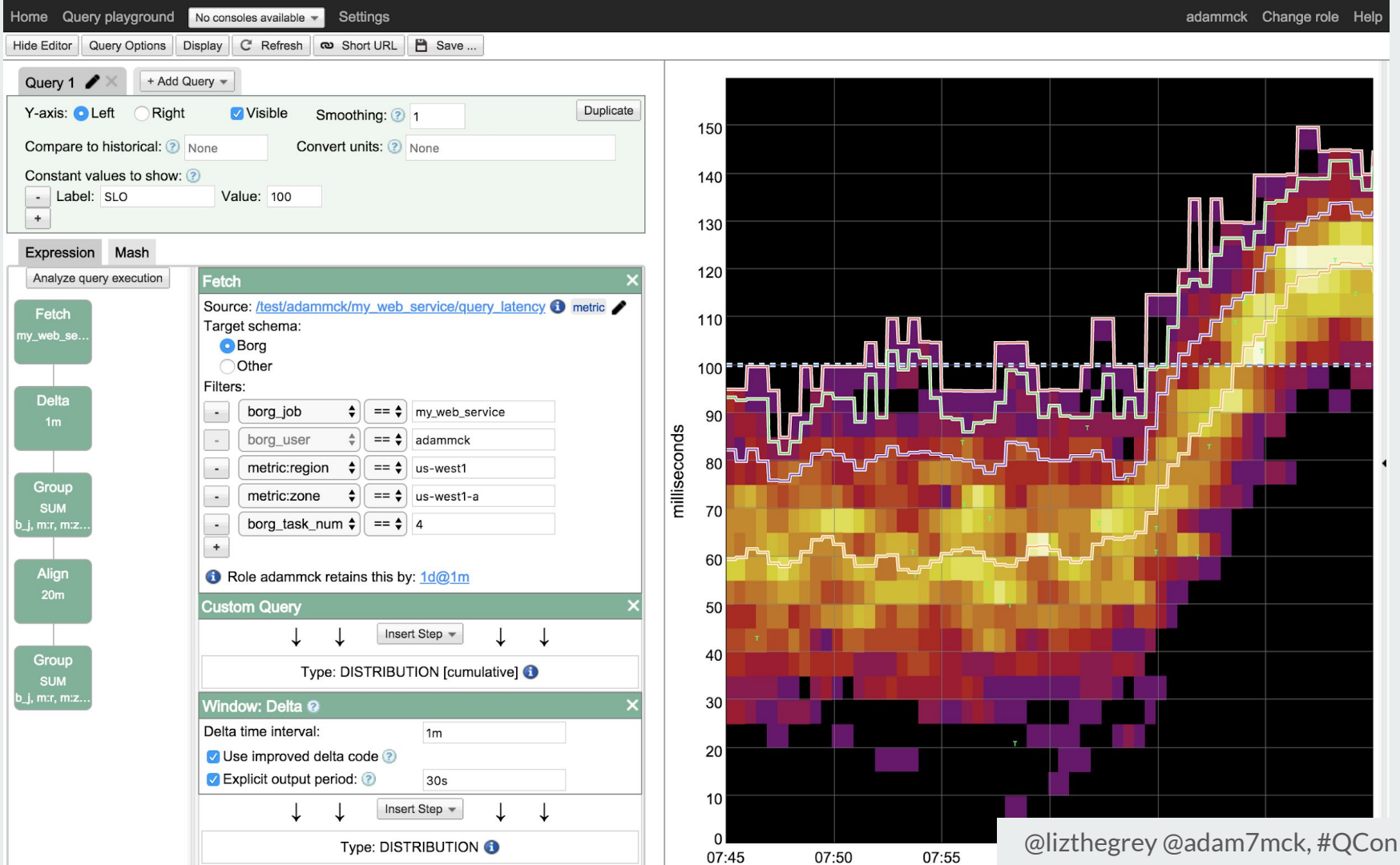
Use improved delta code

Explicit output period: 30s

Insert Step

Type: DISTRIBUTION




Home
Query playground
No consoles available
Settings
adammck
Change role
Help
Hide Editor
Query Options
Display
Refresh
Short URL
Save ...
Query 1
+ Add Query
Y-axis: Left Right Visible
Smoothing: 1
Duplicate
Compare to historical: None
Convert units: None
Constant values to show:
Label: SLO
Value: 100
Expression
Mash
Analyze query execution
Fetch
my_web_se...
Delta
1m
Group
SUM
b_j, m:r, m:z...
Align
20m
Group
SUM
b_j, m:r, m:z...
Fetch
Source: /test/adammck/my_web_service/query_latency
metric
Target schema:
Borg
Other
Filters:
borg_job == my_web_service
borg_user == adammck
metric:region == us-west1
metric:zone == us-west1-a
borg_task_num == 4
Role adammck retains this by: 1d@1m
Custom Query
Insert Step
Type: DISTRIBUTION [cumulative]
Window: Delta
Delta time interval: 1m
Use improved delta code
Explicit output period: 30s
Insert Step
Type: DISTRIBUTION
milliseconds
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
07:45
07:50
07:55

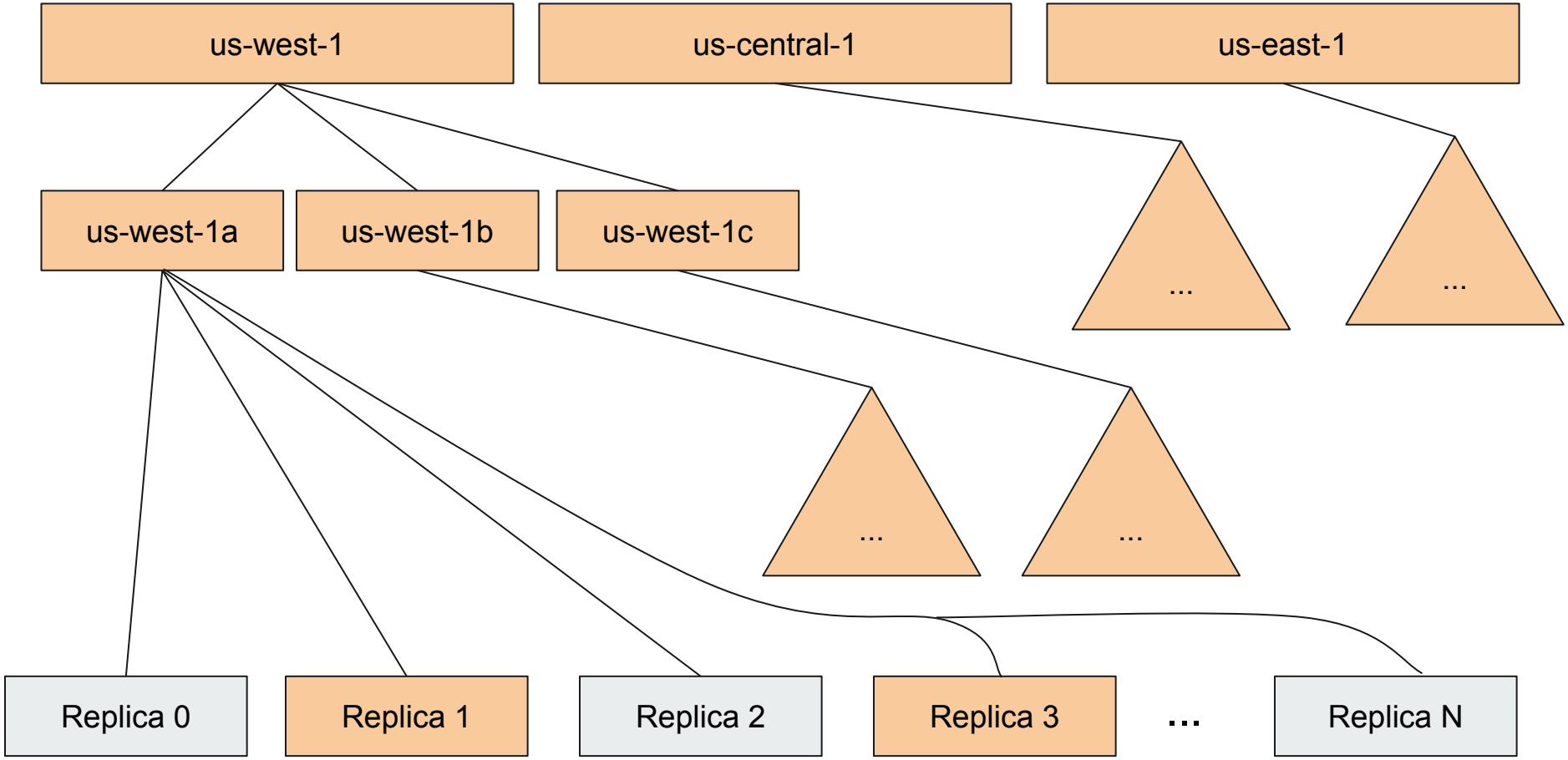
1. Layer Peeling

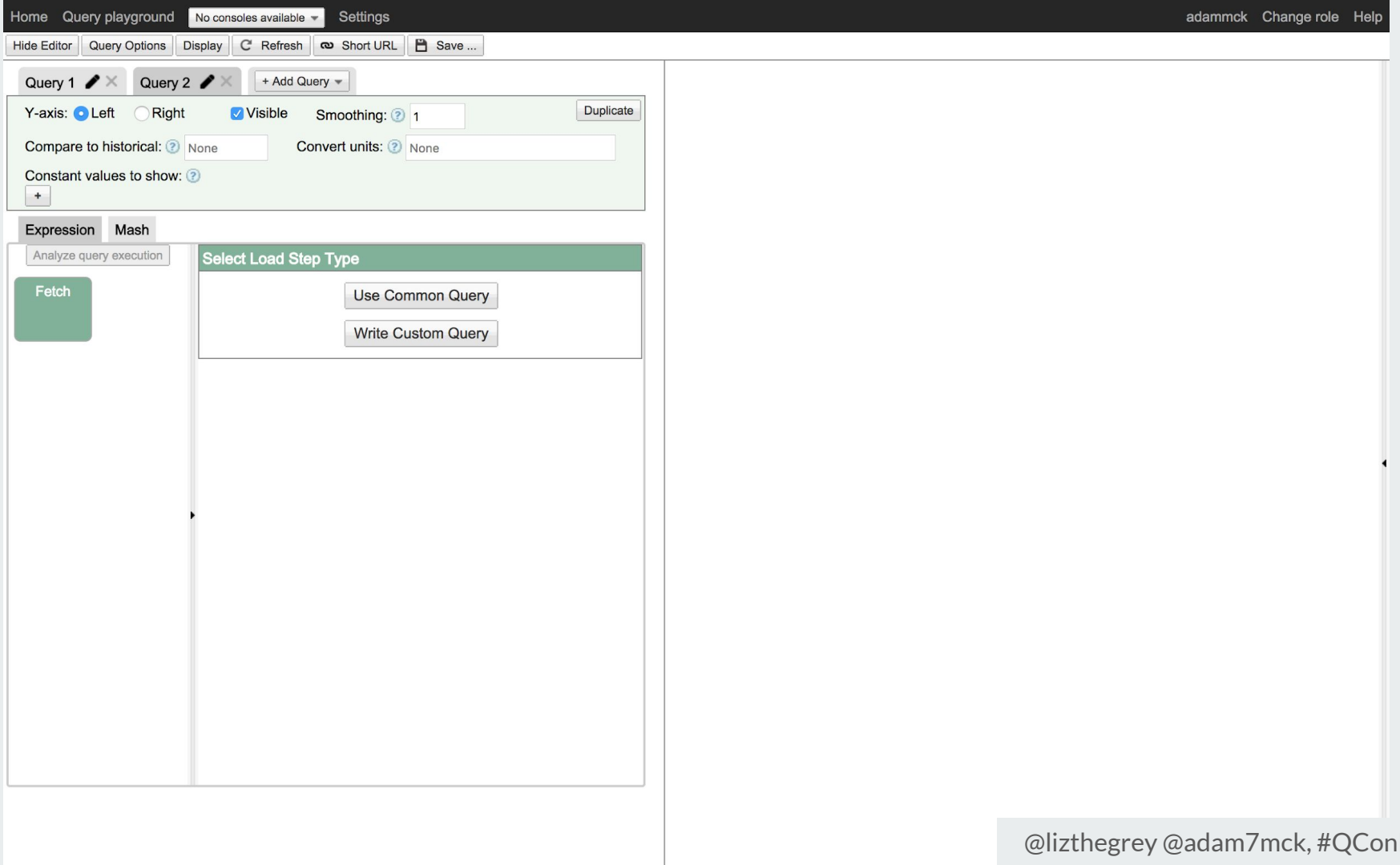
- Reminder: couldn't have plotted all the tasks if cardinality too high
- Instead: repeatedly bisected the problem
- Found a bad task and restarted it!

2. Dynamic Data Joins

2. Dynamic Data Joins

- Tricky problems can’t be diagnosed by looking only at fields of alerting metrics
- We can’t include every possibly-useful piece of information as a metric field
- We need to pull in data from other metrics, and **join it** to find correlations


us-west-1
us-central-1
us-east-1
us-west-1a
us-west-1b
us-west-1c
...
...
...
...
Replica 0
Replica 1
Replica 2
Replica 3
...
Replica N


Home
Query playground
No consoles available
Settings
adammck
Change role
Help
Hide Editor
Query Options
Display
Refresh
Short URL
Save ...
Query 1
Query 2
+ Add Query
Y-axis:
Left
Right
Visible
Smoothing:
1
Duplicate
Compare to historical:
None
Convert units:
None
Constant values to show:
+
Expression
Mash
Analyze query execution
Fetch
Select Load Step Type
Use Common Query
Write Custom Query

Home Query playground No consoles available Settings adammck Change role Help

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

Query 1 | Query 2 | + Add Query

Y-axis: Left Right Visible Smoothing: 1 Duplicate

Compare to historical: None Convert units: None

Constant values to show: +

Expression | Mash

Analyze query execution

Fetch
my_web_se...

Align
20m

Fetch

Source: /test/adammck/my_web_service/kernel_version metric

Target schema:
- Borg
- Other

Filters:
- borg_user == adammck
+

Role adammck retains this by: 1d@1m

Custom Query

Insert Step

Type: LONG

Window: Align

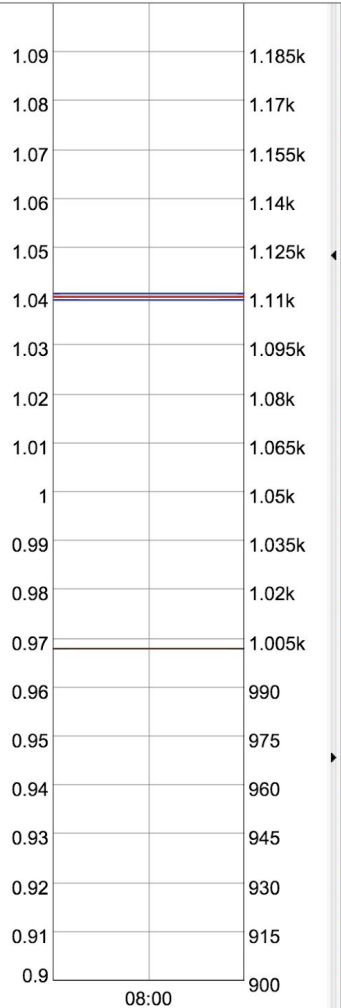
Operator: NEXT_OLDER

Max interpolation gap: 20m

Explicit output period: auto-determined

Insert Step

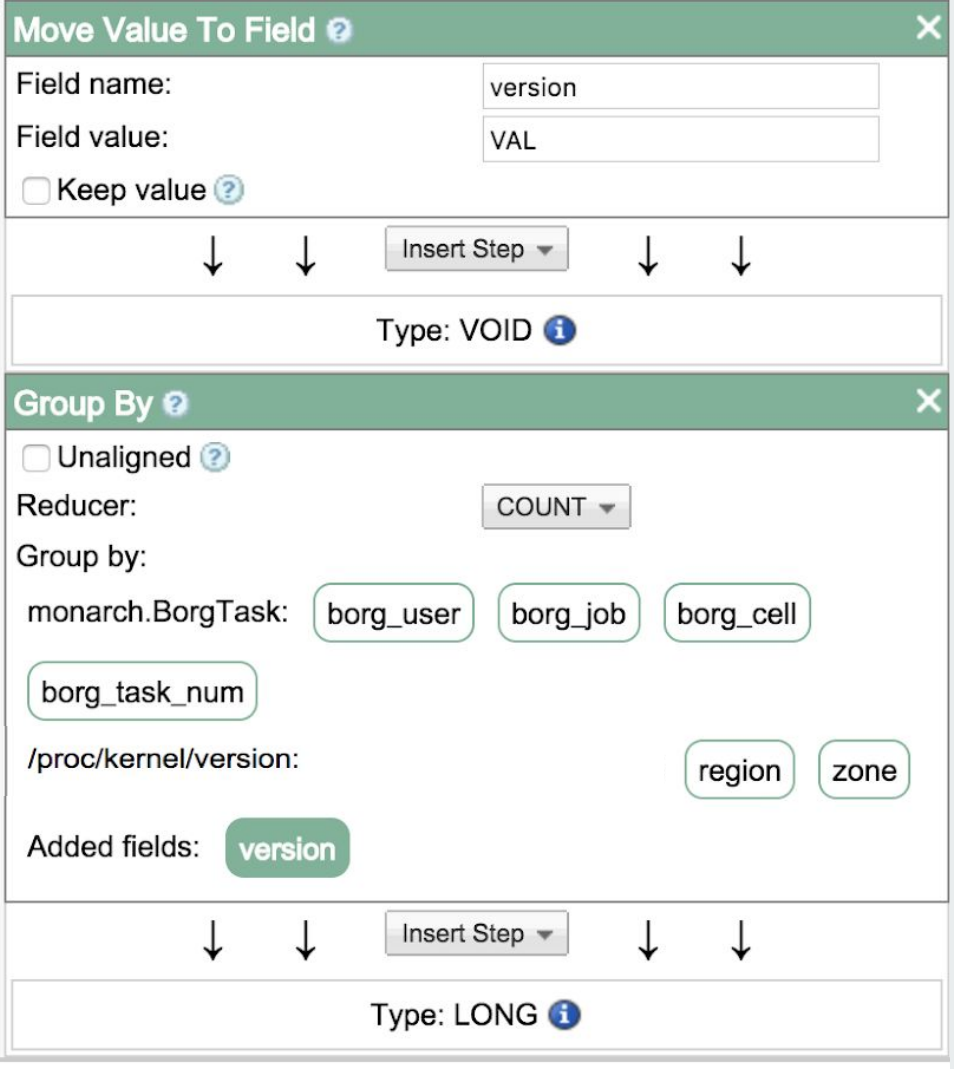
Type: LONG



Query 2

borg_u...	borg_job	borg_cell	borg_t...	region	zone	Latest
adammck	my_web_service	lg	0	us-west1	us-west1-a	1111
adammck	my_web_service	lg	1	us-west1	us-west1-a	1111
adammck	my_web_service	lg	2	us-west1	us-west1-a	1111
adammck	my_web_service	lg	3	us-west1	us-west1-a	1111
adammck	my_web_service	lg	4	us-west1	us-west1-a	1002
adammck	my_web_service	lg	5	us-west1	us-west1-a	1002
adammck	my_web_service	lg	6	us-west1	us-west1-a	1111
adammck	my_web_service	lg	7	us-west1	us-west1-a	1111
adammck	my_web_service	lg	8	us-west1	us-west1-a	1111
adammck	my_web_service	lg	9	us-west1	us-west1-a	1111
adammck	my_web_service	lg	10	us-west1	us-west1-a	1111
adammck	my_web_service	lg	11	us-west1	us-west1-a	1111
adammck	my_web_service	lg	12	us-west1	us-west1-a	1110
adammck	my_web_service	lg	13	us-west1	us-west1-a	1110
adammck	my_web_service	lg	14	us-west1	us-west1-a	1110
adammck	my_web_service	lg	15	us-west1	us-west1-a	1110
adammck	my_web_service	lg	16	us-west1	us-west1-a	1110
adammck						

borg_u…	borg_job	borg_cell	borg_t…	region	zone	Latest
adammck	my_web_service	lg	0	us-west1	us-west1-a	1111


Move Value To Field
Field name:
version
Field value:
VAL
Keep value
Insert Step
Type: VOID
Group By
Unaligned
Reducer:
COUNT
Group by:
monarch.BorgTask:
borg_user
borg_job
borg_cell
borg_task_num
/proc/kernel/version:
region
zone
Added fields:
version
Insert Step
Type: LONG

Home Query playground No consoles available Settings adammck Change role Help

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

Val to Field
version

Group
COUNT
m:v

Role adammck retains this by: 1d@1m

Custom Query

Insert Step

Type: LONG

Window: Align

Operator: NEXT_OLDER

Max interpolation gap: 20m

Explicit output period: auto-determined

Insert Step

Type: LONG

Move Value To Field

Field name: version

Field value: VAL

Keep value

Insert Step

Type: VOID

Group By

Unaligned

Reducer: COUNT

Group by:

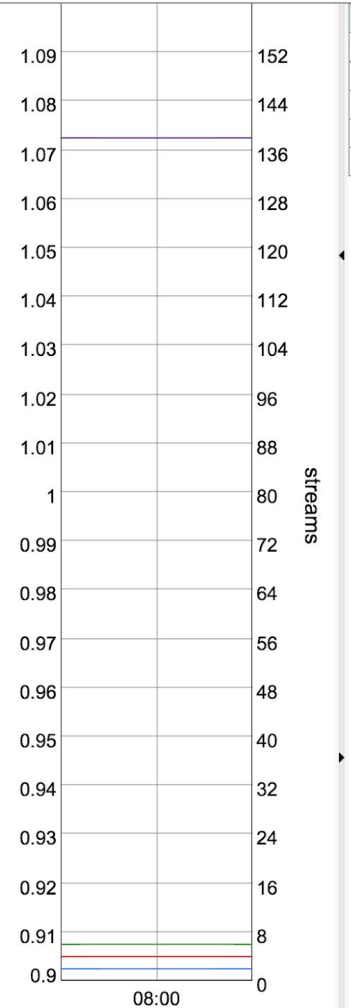
monarch.BorgTask: borg_user borg_job borg_cell borg_task_num

/proc/kernel/version: region zone

Added fields: version

Insert Step

Type: LONG



Query 2	
version	Latest
1002	2
1109	4
1110	6
1111	138

Home Query playground No consoles available Settings adammck Change role Help

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

Query 1 Query 2 + Add Query

Y-axis: Left Right Visible Smoothing: 1 Duplicate

Compare to historical: None Convert units: None

Constant values to show: +

Expression Mash

Analyze query execution

Fetch
my_web_se...

Delta
1m

Group
SUM
m:r, m:z, b_...

Point
Percentile(9...

Fetch

Source: /test/adammck/my_web_service/query_latency metric

Target schema:
- Borg
- Other

Filters:
- borg_job == my_web_service
- borg_user == adammck
+

Role adammck retains this by: 1d@1m

Custom Query

Insert Step

Type: DISTRIBUTION [cumulative]

Window: Delta

Delta time interval: 1m

Use improved delta code

Explicit output period: auto-determined

Insert Step

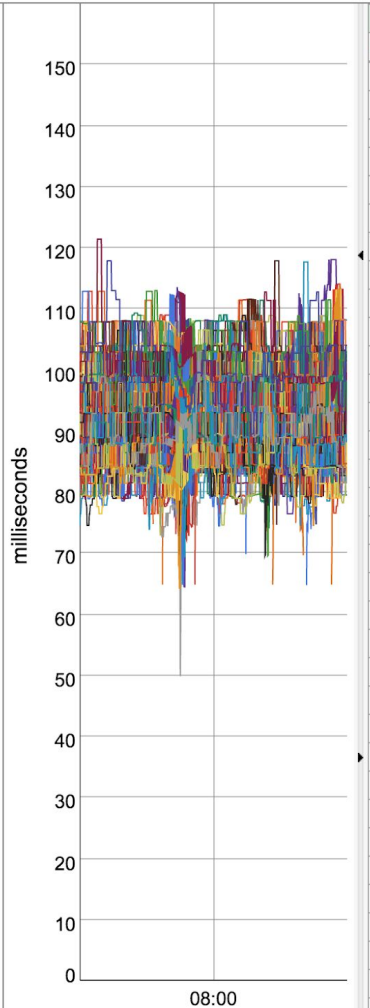
Type: DISTRIBUTION

Group By

Unaligned

Reducer: SUM

Sample as exemplars



Query 1

borg_task_num	region	zone	Latest
0	us-central1	us-central1-a	98.200
0	us-east1	us-east1-a	88.200
0	us-west1	us-west1-a	84.550
0	us-west1	us-west1-b	91.600
0	us-west1	us-west1-c	101.600
1	us-central1	us-central1-a	93.200
1	us-east1	us-east1-a	83.640
1	us-west1	us-west1-a	91.550
1	us-west1	us-west1-b	86.550
1	us-west1	us-west1-c	96.600
2	us-central1	us-central1-a	93.200
2	us-east1	us-east1-a	83.100
2	us-west1	us-west1-a	92.733
2	us-west1	us-west1-b	98.200
2	us-west1	us-west1-c	98.100
3	us-central1	us-central1-a	87.700
3	us-east1	us-east1-a	96.600
3	us-west1	us-west1-a	94.367
3	us-west1	us-west1-b	91.550
3	us-west1	us-west1-c	98.100
4	us-central1	us-central1-a	93.200
4	us-east1	us-east1-a	81.600
4	us-west1	us-west1-a	141.550
4	us-west1	us-west1-b	103.200
4	us-west1	us-west1-c	104.100
5	us-central1	us-central1-a	93.200
5	us-east1	us-east1-a	83.200
5	us-west1	us-west1-a	139.100
5	us-west1	us-west1-b	88.200
5	us-west1	us-west1-c	102.733
6	us-central1	us-central1-a	88.300

Home Query playground No consoles available Settings adammck Change role Help

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

Query 1 Query 2 Query 3 + Add Query

Y-axis: Left Right Visible Smoothing: 1 Duplicate

Compare to historical: None Convert units: None

Constant values to show: +

Expression Mash

Analyze query execution

Fetch my_web_se... → Delta 1m → Group SUM m:r, m:z, b_... → Point Percentile(9...

Fetch my_web_se... → Align 20m → Group PICK_ANY m:r, m:z, b_... → Val to Field version

Join latency kernel_version

Fetch

Source: /test/adammck/my_web_service/kernel_version metric

Target schema: Borg Other

Filters: borg_user == adammck

Role adammck retains this by: 1d@1m

Custom Query

Insert Step

Type: LONG

Window: Align

Operator: NEXT_OLDER

Max interpolation gap: 20m

Explicit output period: auto-determined

Insert Step

Type: LONG

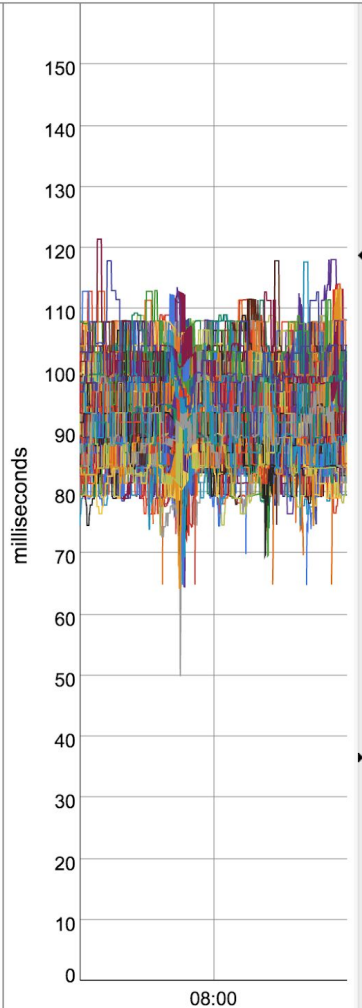
Group By

Unaligned

Reducer: PICK_ANY

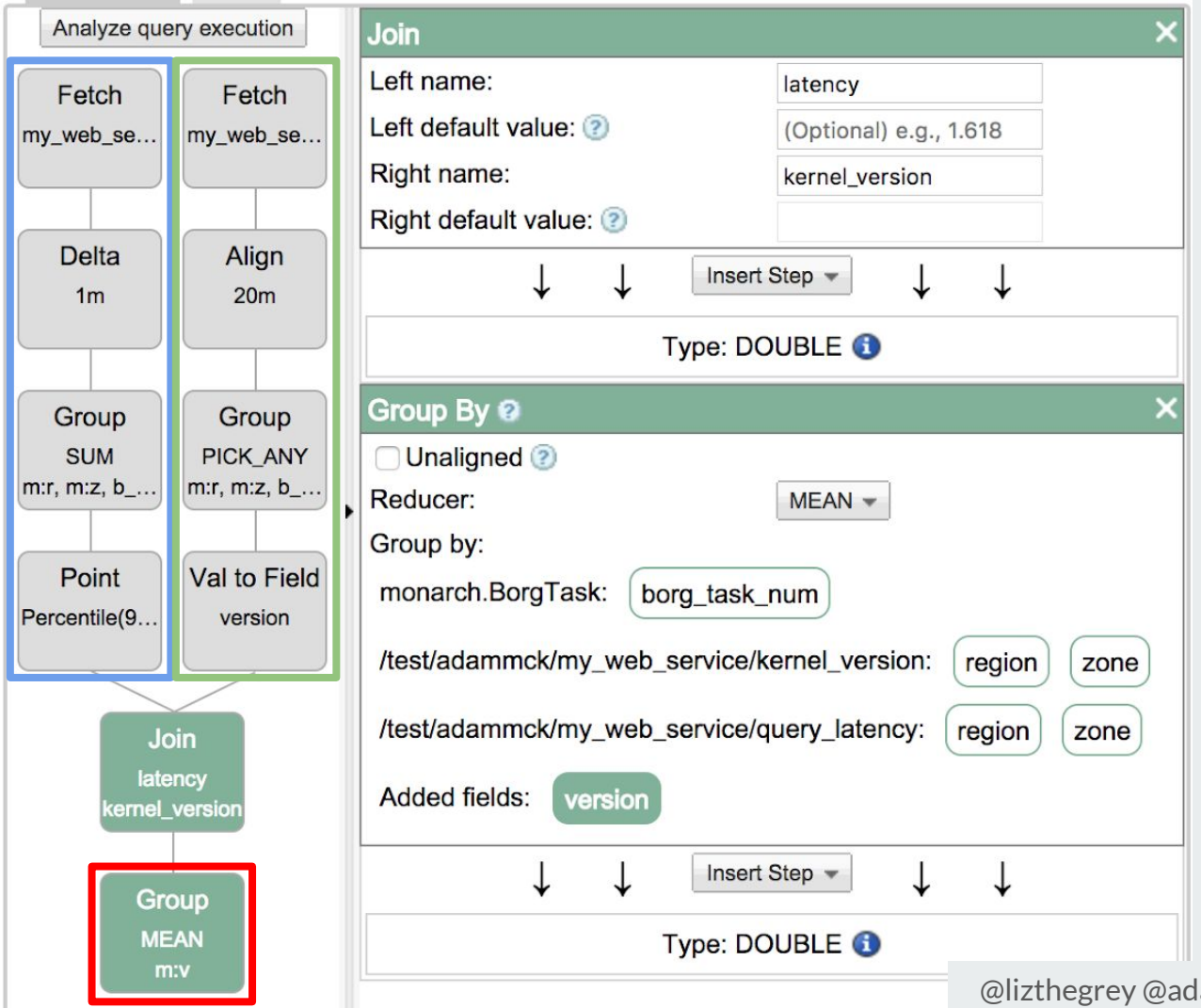
Group by:

monarch.BorgTask: borg_user borg_job borg_cell



Query 3

borg_task_...	region	version	zone	Latest
0	us-central1	1111	us-central1-a	98.200
0	us-east1	1111	us-east1-a	88.200
0	us-west1	1111	us-west1-a	84.550
0	us-west1	1111	us-west1-b	91.600
0	us-west1	1111	us-west1-c	101.600
1	us-central1	1111	us-central1-a	93.200
1	us-east1	1111	us-east1-a	83.640
1	us-west1	1111	us-west1-a	91.550
1	us-west1	1111	us-west1-b	86.550
1	us-west1	1111	us-west1-c	96.600
2	us-central1	1111	us-central1-a	93.200
2	us-east1	1111	us-east1-a	83.100
2	us-west1	1111	us-west1-a	92.733
2	us-west1	1111	us-west1-b	98.200
2	us-west1	1111	us-west1-c	98.100
3	us-central1	1111	us-central1-a	87.700
3	us-east1	1111	us-east1-a	96.600
3	us-west1	1111	us-west1-a	94.367
3	us-west1	1111	us-west1-b	91.550
3	us-west1	1111	us-west1-c	98.100
4	us-central1	1111	us-central1-a	93.200
4	us-east1	1111	us-east1-a	81.600
4	us-west1	1002	us-west1-a	141.550
4	us-west1	1111	us-west1-b	103.200
4	us-west1	1111	us-west1-c	104.100
5	us-central1	1111	us-central1-a	93.200
5	us-east1	1111	us-east1-a	83.200
5	us-west1	1002	us-west1-a	139.100
5	us-west1	1111	us-west1-b	88.200
5	us-west1	1111	us-west1-c	102.733
6	us-central1	1111	us-central1-a	88.300


Analyze query execution
Fetch
my_web_se…
Delta
1m
Group
SUM
m:r, m:z, b_…
Point
Percentile(9…
Fetch
my_web_se…
Align
20m
Group
PICK_ANY
m:r, m:z, b_…
Val to Field
version
Join
latency
kernel_version
Group
MEAN
m:v
Join
Left name:
latency
Left default value:
(Optional) e.g., 1.618
Right name:
kernel_version
Right default value:
Insert Step
Type: DOUBLE
Group By
Unaligned
Reducer:
MEAN
Group by:
monarch.BorgTask:
borg_task_num
/test/adammck/my_web_service/kernel_version:
region
zone
/test/adammck/my_web_service/query_latency:
region
zone
Added fields:
version
Insert Step
Type: DOUBLE

Home Query playground No consoles available Settings adammck Change role Help

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

Query 1 Query 2 Query 3 + Add Query

Y-axis: Left Right Visible Smoothing: 1 Duplicate

Compare to historical: None Convert units: None

Constant values to show: +

Expression Mash

Analyze query execution

Fetch my_web_se... Fetch my_web_se...

Delta 1m Align 20m

Group SUM m:r, m:z, b_... Group PICK_ANY m:r, m:z, b_...

Point Percentile(9... Val to Field version

Join latency kernel_version

Group MEAN m:v

Join

Left name: latency

Left default value: (Optional) e.g., 1.618

Right name: kernel_version

Right default value:

Insert Step

Type: DOUBLE

Group By

Unaligned

Reducer: MEAN

Group by:

monarch.BorgTask: borg_task_num

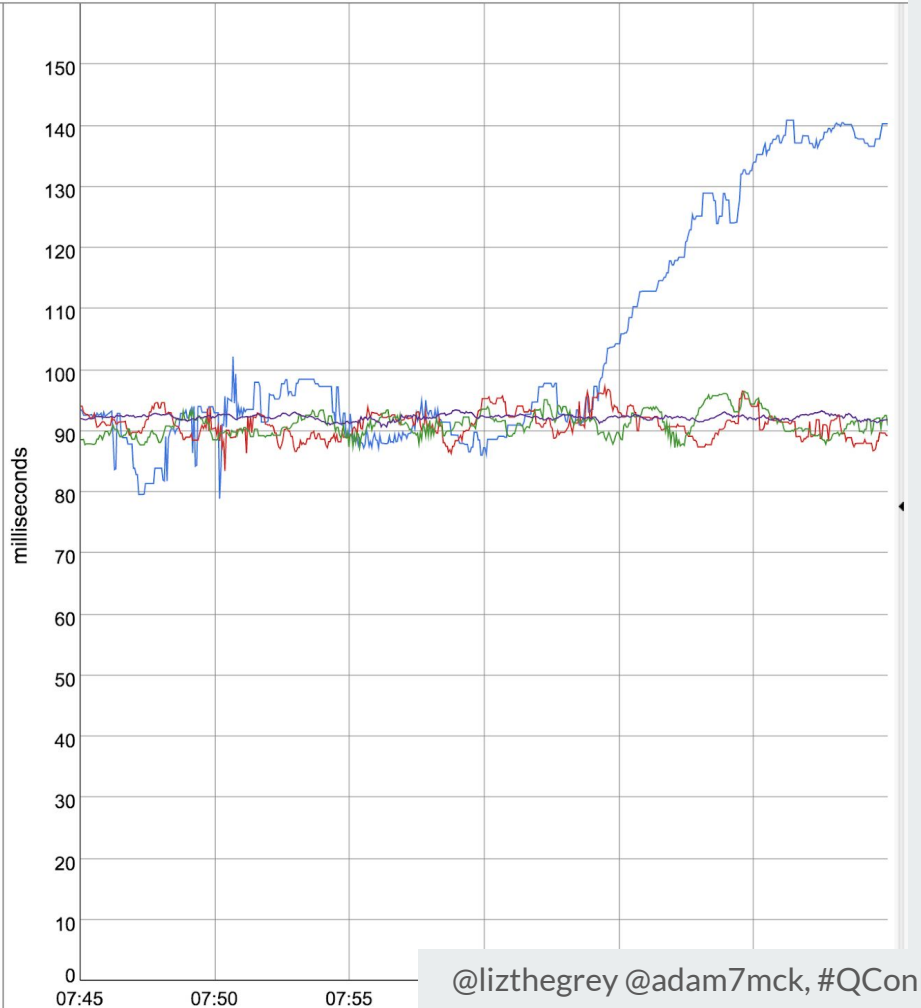
/test/adammck/my_web_service/kernel_version: region zone

/test/adammck/my_web_service/query_latency: region zone

Added fields: version

Insert Step

Type: DOUBLE



Home Query playground No consoles available Settings adammck Change role Help

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

Query 1 | Query 2 | Query 3 | + Add Query

Y-axis: Left Right Visible Smoothing: 1 Duplicate

Compare to historical: None Convert units: None

Constant values to show: +

Expression | Mash

Analyze query execution

Fetch my_web_se... | Fetch my_web_se...

Delta 1m | Align 20m

Group SUM m:r, m:z, b_... | Group PICK_ANY m:r, m:z, b_...

Point Percentile(9... | Val to Field version

Join latency kernel_version

Group MEAN m:v

Join

Left name: latency

Left default value: (Optional) e.g., 1.618

Right name: kernel_version

Right default value:

Insert Step

Type: DOUBLE

Group By

Unaligned

Reducer: MEAN

Group by:

monarch.BorgTask: borg_task_num

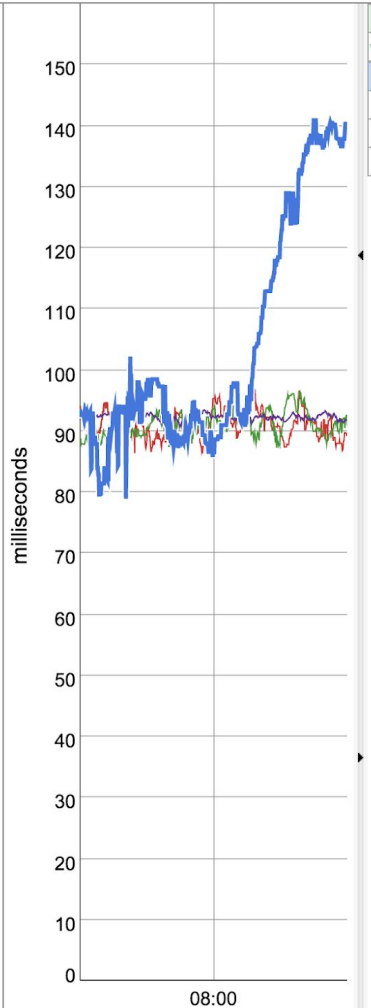
/test/adammck/my_web_service/kernel_version: region zone

/test/adammck/my_web_service/query_latency: region zone

Added fields: version

Insert Step

Type: DOUBLE



Query 3	
version	Latest
1002	140.325
1109	89.300
1110	90.908
1111	91.924

2. Dynamic Data Joins

- Tricky problems can’t be diagnosed by looking only at fields of alerting metrics
- We can’t include every possibly-useful piece of information as a metric field
- We need to pull in data from other metrics, and **join it** to find correlations

3. Exemplars

3. Exemplars

Make distributed tracing & high cardinality streams *actually useful*.

Sample (1) high-cardinality tags and/or (2) attached traces to pinpoint issues

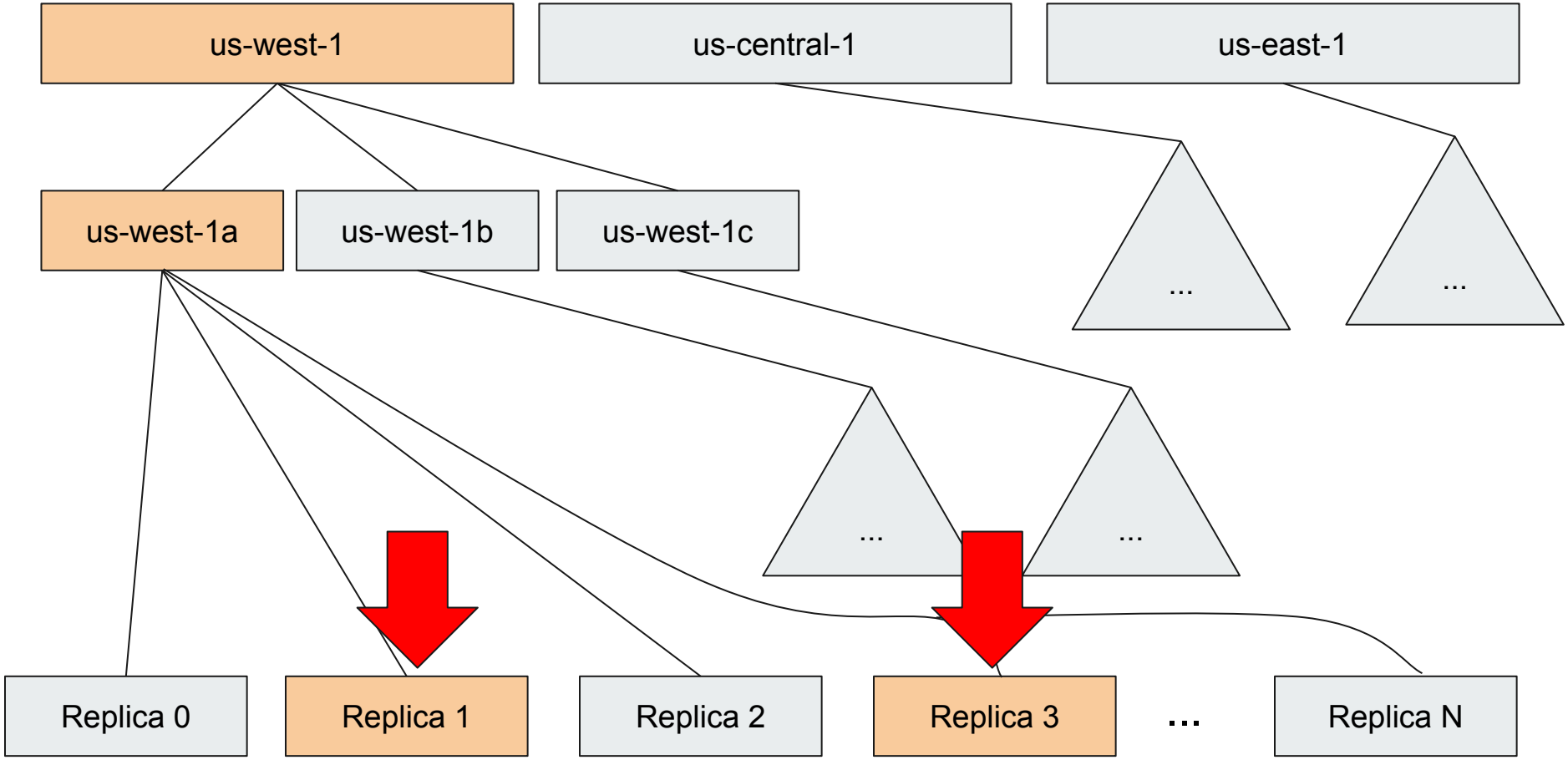
Probabilistically keep up to 1 per (time bucket x value histogram bucket)

Mark in your graphing tool of choice (e.g. Grafana) which buckets have samples

Highlight samples sharing same tags; link to fine-grained info (e.g. Zipkin traces)

Exemplar Tags

What if we could skip all those messy layer peeling steps and get directly to bad tasks?


us-west-1
us-central-1
us-east-1
us-west-1a
us-west-1b
us-west-1c
...
...
...
...
Replica 0
Replica 1
Replica 2
Replica 3
...
Replica N

Home Query playground No consoles available Settings adammck Change role Help

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

Query 1 + Add Query

Y-axis: Left Right Visible Smoothing: 1 Duplicate

Compare to historical: None Convert units: None

Constant values to show: +

Expression Mash

Analyze query execution

Fetch my_web_se...

Delta 1m

Group SUM b_j, m:r, m:z...

Align 20m

Group SUM

Align 20m

Fetch

Source: /test/adammck/my_web_service/query_latency metric

Target schema:

Borg

Other

Filters:

- borg_job == my_web_service
- borg_user == adammck
+

Role adammck retains this by: 1d@1m

Custom Query

Insert Step

Type: DISTRIBUTION [cumulative]

Window: Delta

Delta time interval: 1m

Use improved delta code

Explicit output period: 1m

Insert Step

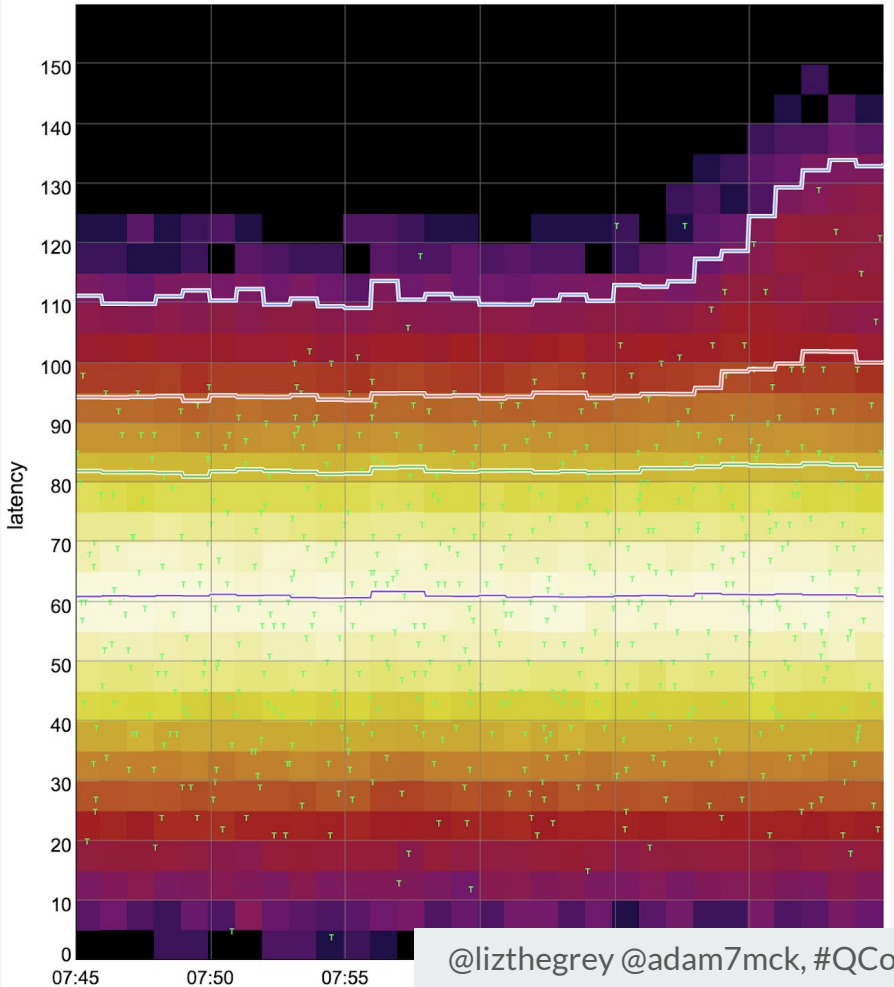
Type: DISTRIBUTION

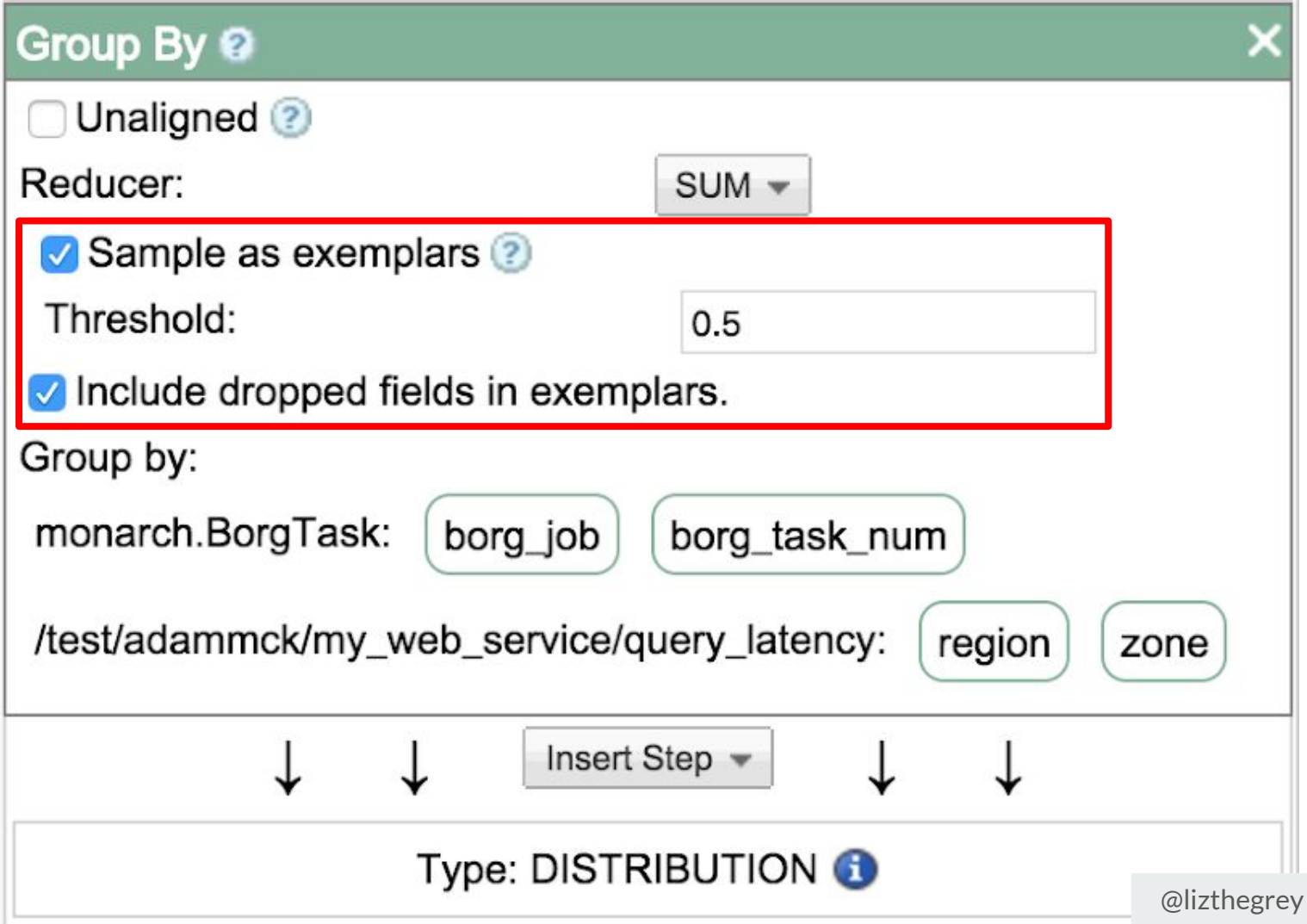
Group By

Unaligned

Reducer: SUM

Sample as exemplars




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

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

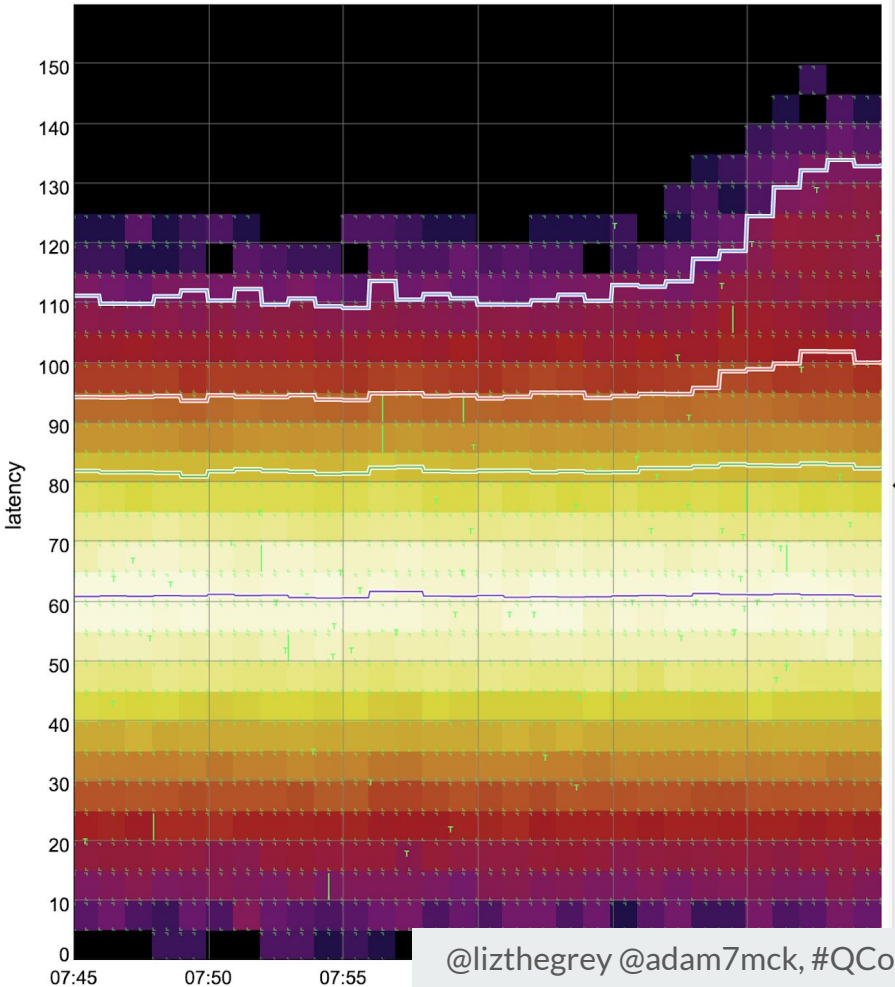
Operator: NEXT_OLDER

Max interpolation gap: 20m

Explicit output period: auto-determined

Insert Step

Type: DISTRIBUTION



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

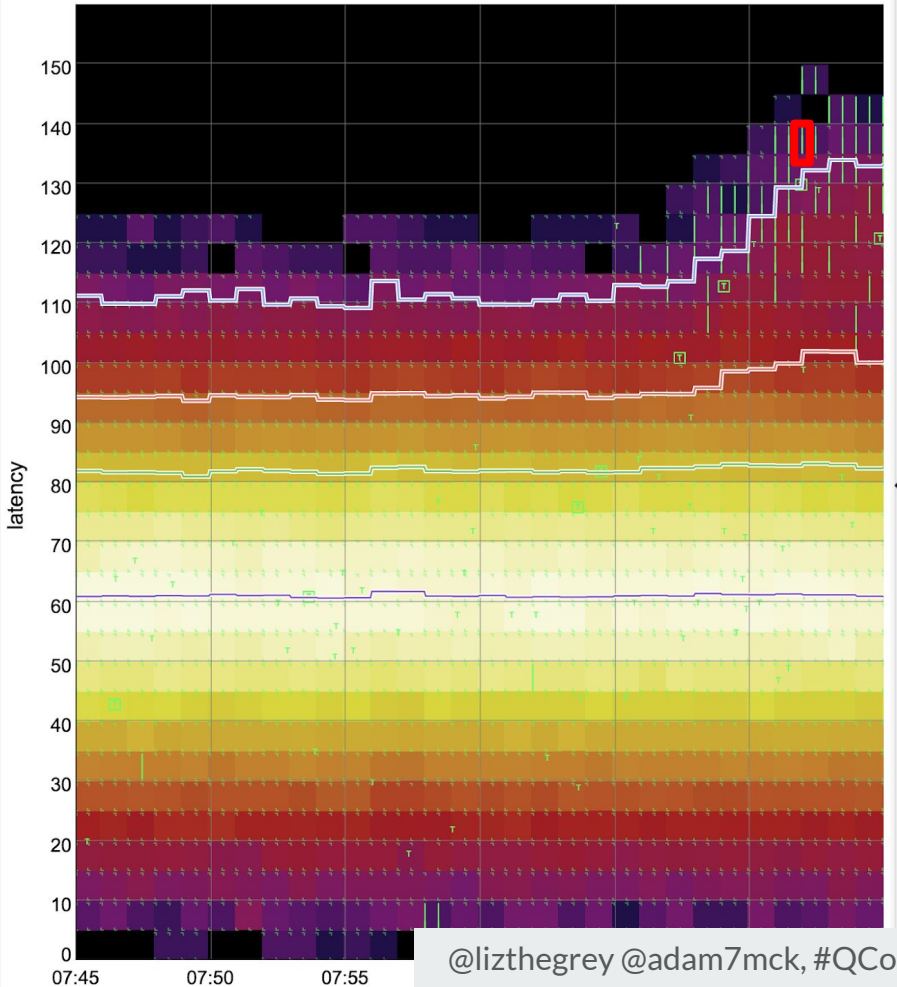
Operator: NEXT_OLDER

Max interpolation gap: 20m

Explicit output period: auto-determined

Insert Step

Type: DISTRIBUTION



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

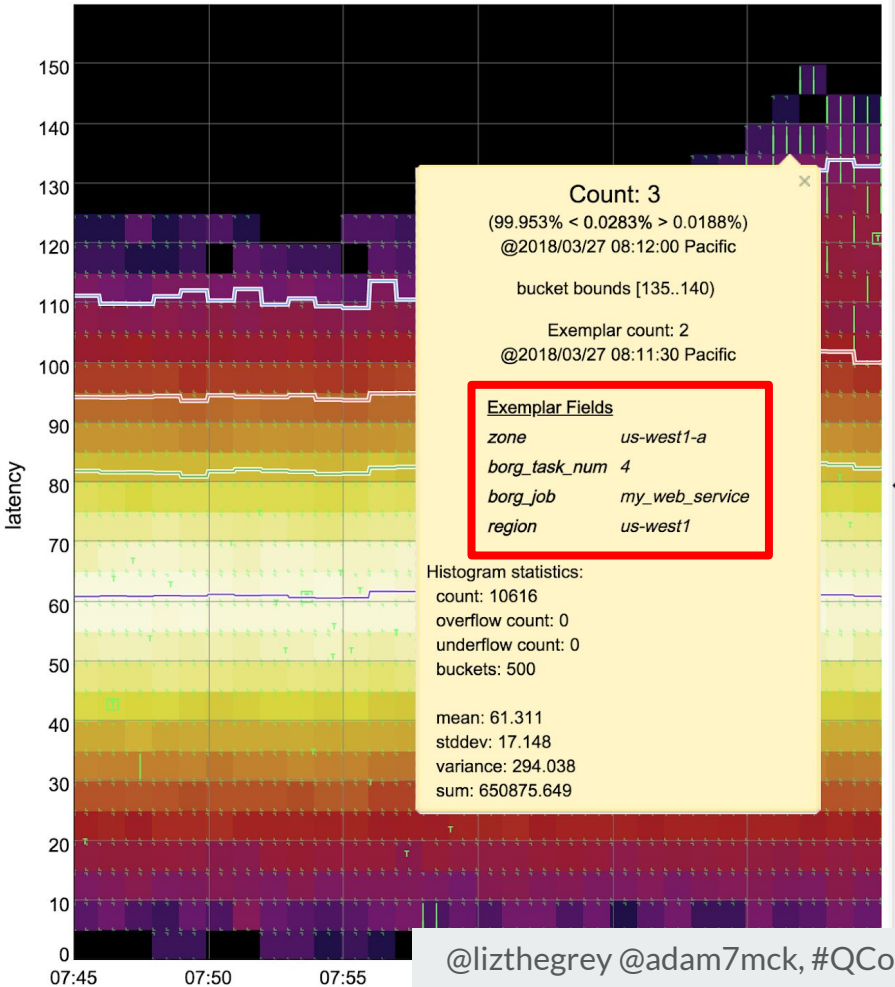
Operator: NEXT_OLDER

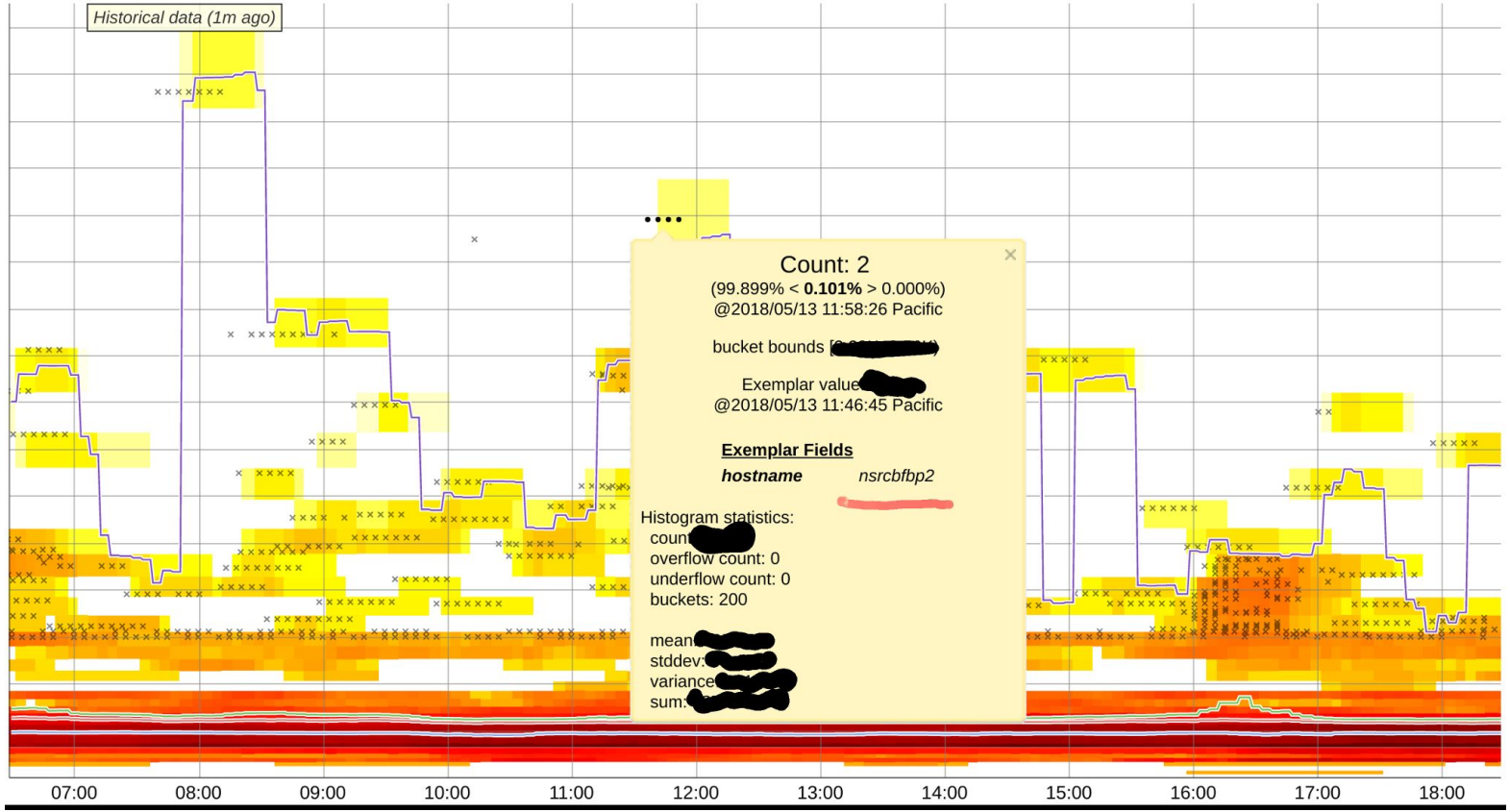
Max interpolation gap: 20m

Explicit output period: auto-determined

Insert Step

Type: DISTRIBUTION





Real world example: Machine Boot Times

Exemplar Traces

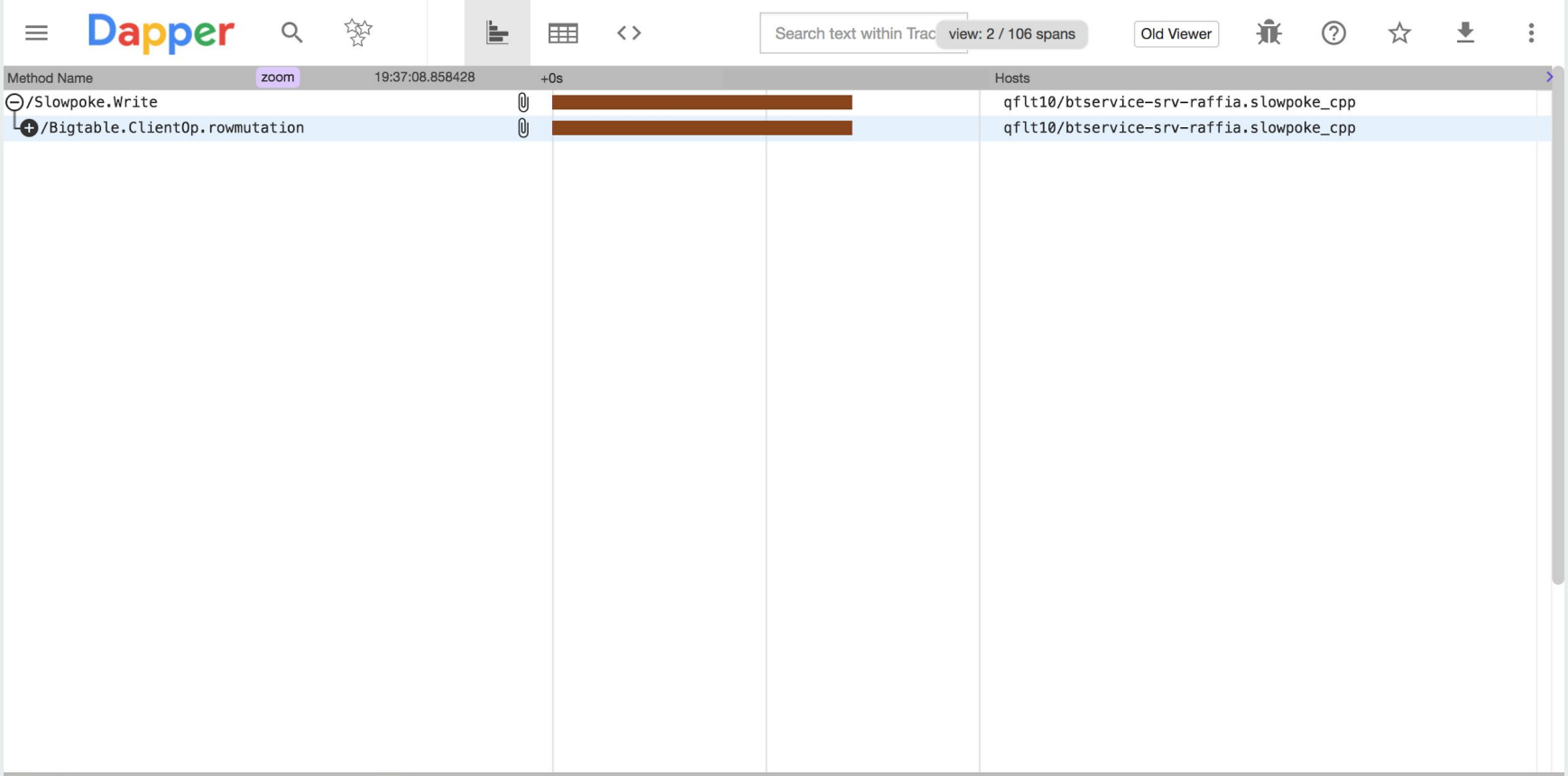
These are real graphs from our blackbox monitoring of a storage system at Google.

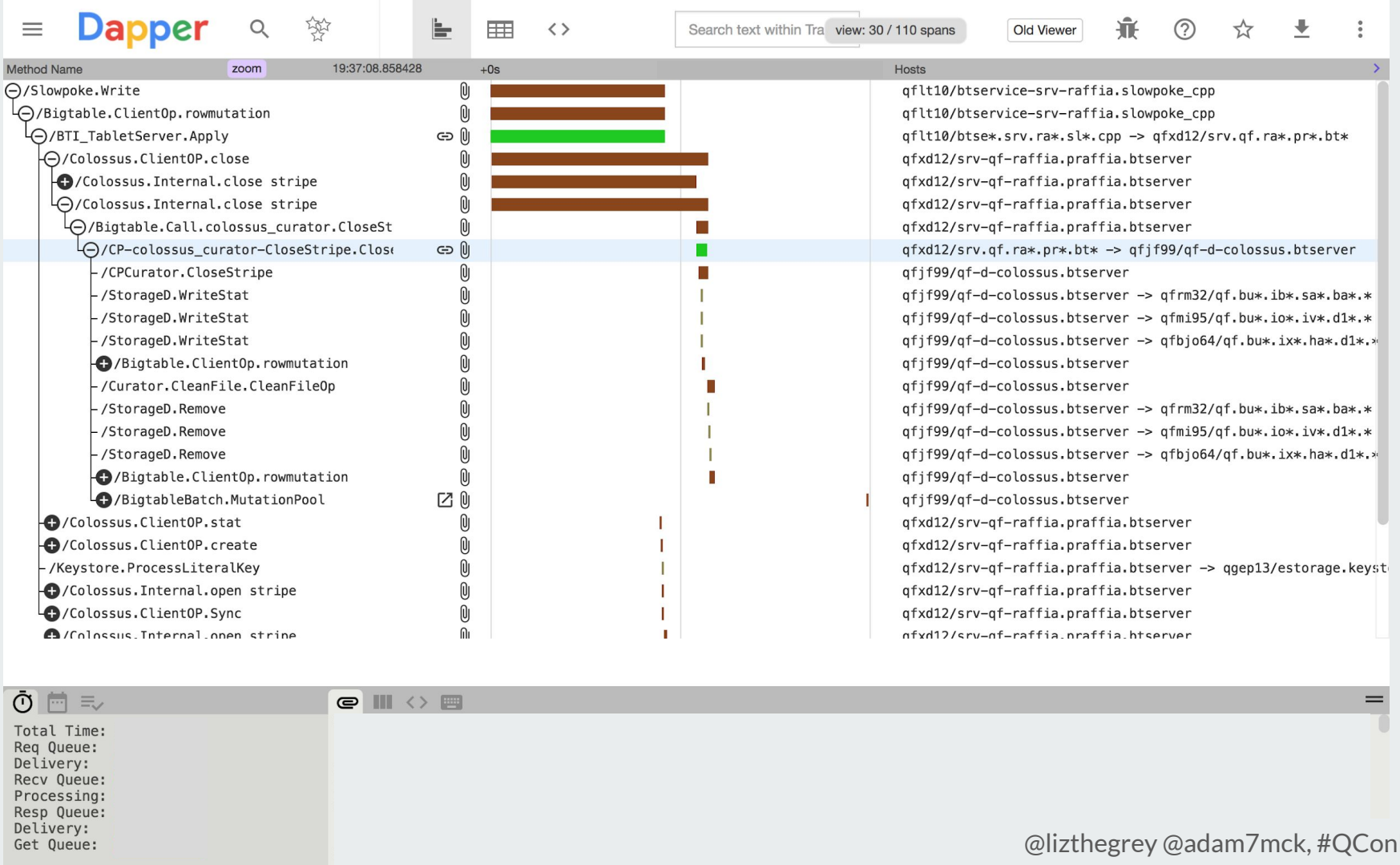
We've instrumented the blackbox prober to sample traces at a high rate (~1%).

Blackbox prober latency is an SLI for the system.

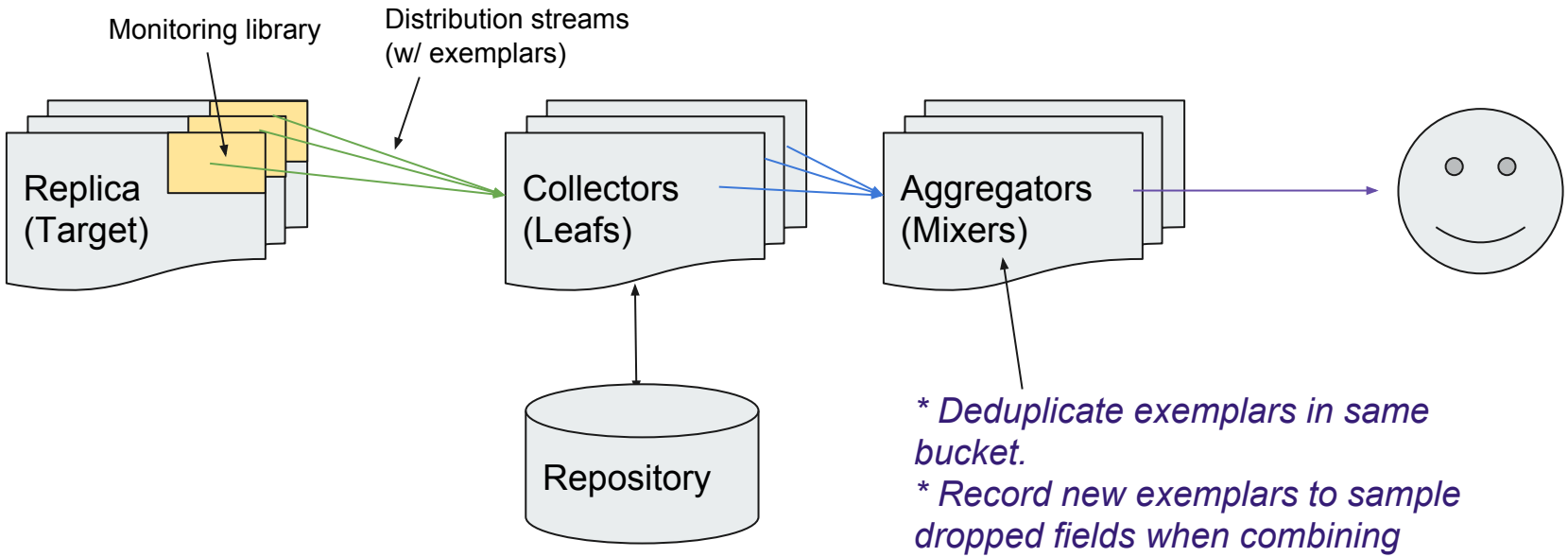
Query playground
Select console
Settings
bt-sre
Change role
Editor
Query Options
Display
Refresh
Short URL
Save ...
write (
18:20
18:30
18:40
18:50
19:00
19:10
19:20
19:30

Query playground
Select console
Settings
bt-sre
Change role
Editor
Query Options
Display
Refresh
Short URL
Save ...
write (
Count: 7
(99.330% < 0.391% > 0.279%)
@2018/01/11 19:42:44 Pacific
bucket bounds [ )
Exemplar value:
@2018/01/11 19:41:44 Pacific
Dapper Trace
Exemplar Fields
Histogram statistics:
count: 1792
overflow count: 0
underflow count: 0
buckets: 117
mean: 34.242
stddev: 117.498
variance: 13805.785
sum: 61360.995
18:20
18:30
18:40
18:50
19:00
19:10
19:20
19:30


Dapper
Search text within Trac
view: 2 / 106 spans
Old Viewer
Method Name
zoom
19:37:08.858428
+0s
Hosts
/Slowpoke.Write
qflt10/btservice-srv-raffia.slowpoke_cpp
/Bigtable.ClientOp.rowmutation
qflt10/btservice-srv-raffia.slowpoke_cpp
Total Time:
Req Queue:
Delivery:
Recv Queue:
Processing:
Resp Queue:
Delivery:
Get Queue:


Dapper
Search text within Tra
view: 30 / 110 spans
Old Viewer
Method Name
zoom
19:37:08.858428
+0s
Hosts
/Slowpoke.Write
/Bigtable.ClientOp.rowmutation
/BTI_TabletServer.Apply
/Colossus.ClientOP.close
/Colossus.Internal.close stripe
/Colossus.Internal.close stripe
/Bigtable.Call.colossus_curator.CloseSt
/CP-colossus_curator-CloseStripe.Clos
/CPCurator.CloseStripe
/StorageD.WriteStat
/StorageD.WriteStat
/StorageD.WriteStat
/Bigtable.ClientOp.rowmutation
/Curator.CleanFile.CleanFileOp
/StorageD.Remove
/StorageD.Remove
/StorageD.Remove
/Bigtable.ClientOp.rowmutation
/BigtableBatch.MutationPool
/Colossus.ClientOP.stat
/Colossus.ClientOP.create
/Keystore.ProcessLiteralKey
/Colossus.Internal.open stripe
/Colossus.ClientOP.Sync
qflt10/btservice-srv-raffia.slowpoke_cpp
qflt10/btservice-srv-raffia.slowpoke_cpp
qflt10/btse*.srv.ra*.sl*.cpp -> qfxd12/srv.qf.ra*.pr*.bt*
qfxd12/srv-qf-raffia.praffia.btserver
qfxd12/srv-qf-raffia.praffia.btserver
qfxd12/srv-qf-raffia.praffia.btserver
qfxd12/srv-qf-raffia.praffia.btserver
qfxd12/srv.qf.ra*.pr*.bt* -> qfjf99/qf-d-colossus.btserver
qfjf99/qf-d-colossus.btserver
qfjf99/qf-d-colossus.btserver -> qfrm32/qf.bu*.ib*.sa*.ba*.*
qfjf99/qf-d-colossus.btserver -> qfmi95/qf.bu*.io*.iv*.d1*.*
qfjf99/qf-d-colossus.btserver -> qfbjo64/qf.bu*.ix*.ha*.d1*.*
qfjf99/qf-d-colossus.btserver
qfjf99/qf-d-colossus.btserver
qfjf99/qf-d-colossus.btserver -> qfrm32/qf.bu*.ib*.sa*.ba*.*
qfjf99/qf-d-colossus.btserver -> qfmi95/qf.bu*.io*.iv*.d1*.*
qfjf99/qf-d-colossus.btserver -> qfbjo64/qf.bu*.ix*.ha*.d1*.*
qfjf99/qf-d-colossus.btserver
qfjf99/qf-d-colossus.btserver
qfxd12/srv-qf-raffia.praffia.btserver
qfxd12/srv-qf-raffia.praffia.btserver
qfxd12/srv-qf-raffia.praffia.btserver -> qgep13/estorage.keyst
qfxd12/srv-qf-raffia.praffia.btserver
qfxd12/srv-qf-raffia.praffia.btserver
Total Time:
Req Queue:
Delivery:
Recv Queue:
Processing:
Resp Queue:
Delivery:
Get Queue:

System Diagram



** Deduplicate exemplars in same bucket.*
** Record new exemplars to sample dropped fields when combining streams*

Exemplar Trace Collection Code

```
func (d *Distribution) AddValueToDistribution(v float64, ctx context.Context) {
  t := time.Now()
  vBucket := d.BucketV(v)
  tBucket := d.BucketT(t)
  d.IncrementCounter(vBucket, tBucket)

  if d.CollectExemplars && ctx.Trace != nil {
    if !d.HasExemplar(vBucket, tBucket) {
      d.PutExemplar(vBucket, tBucket, v, t, ctx.Trace.SpanId, ctx.Trace.TraceId)
    }
  }
}
```

Exemplar Client-side Tag Collection Code

```
func (d *Distribution) AddValueToDistribution(v float64, ctx context.Context) {
  t := time.Now()
  vBucket := d.BucketV(v)
  tBucket := d.BucketT(t)
  d.IncrementCounter(vBucket, tBucket)

  if d.CollectExemplars && ctx.CustomTag != nil {  // per-req non-PII eg video id
    if !d.HasExemplar(vBucket, tBucket) {
      d.PutExemplar(vBucket, tBucket, v, t, ctx.CustomTag)
    }
  }
}
```

Exemplar Deduplication Code

```
func (a *Distribution) MergeDistribution(d []*Distribution) {
  exemplars := make(map[BucketVT][]Exemplar)
  for _, dist := range d {
    for _, bucket := range dist.Buckets() {
      a.AddBucketCounts(bucket)
      if bucket.Exemplar != nil {
        exemplars[bucket.VT()] := append(exemplars[bucket.VT()], bucket.Exemplar)
      }
    }
  }
  for slot, list := range exemplars {
    a.PutExemplar(slot, list[random.Intn(len(list))])
  }
}
```

Exemplar Cardinality Sampling Code

```
func (a *Distribution) MergeDistribution(d []*Distribution) {
  exemplars := make(map[BucketVT][]Exemplar)
  for _, dist := range d {
    for _, bucket := range dist.Buckets() {
      a.AddBucketCounts(bucket)
      exemplars[bucket.VT()] := append(exemplars[bucket.VT()], dist.AllTags)
    }
  }
  for slot, list := range exemplars {
    a.PutExemplar(slot, weighted_random_choice(list))
  }
}
```

3. Exemplars

With traces (or culprit tags) in hand, easier to form hypotheses

Just knowing which host and timestamp to look at helps (e.g. logs, local stack traces)

Testing hypotheses still needs dynamic query building & evaluation

Hypothesize -> test gets faster!

Dtracing, distribution graphing exist in wild; traces as exemplars are being adopted!

Conclusion

How did we improve resiliency?

Formulated hypotheses faster and tested them faster

Our 3 best practices:

1. **Layer Peeling:** precomputation and interactive drilldowns
2. **Dynamic Data Joins:** use fewer pre-specified tags
3. **Exemplars:** dtracing + distributions

Post-incident cleanup

Add critical queries for debugging into playbook

Evaluate compute, storage, & cognitive costs of adding precomputes and dashboards

No two outages are identical

Summary

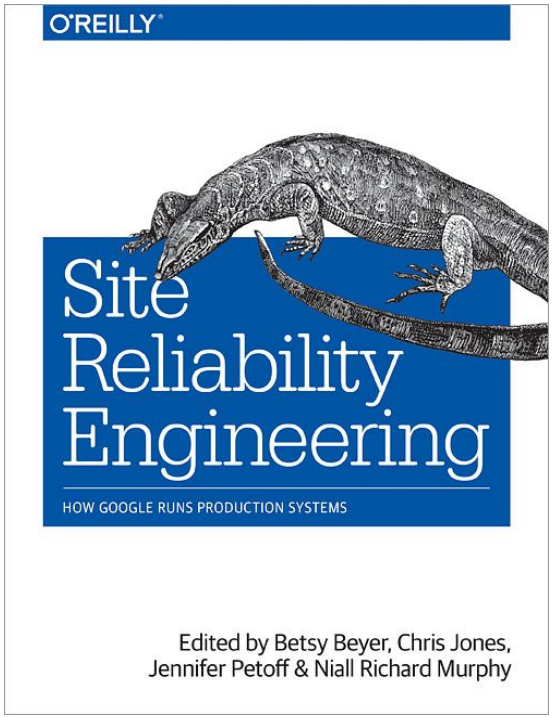
- Reduced time to resolve = less error budget spend
- Ad-hoc querying and drilling down [not 100+ dashboards] are the future
- If you can't quickly and confidently root-cause and test fixes, you're in pain

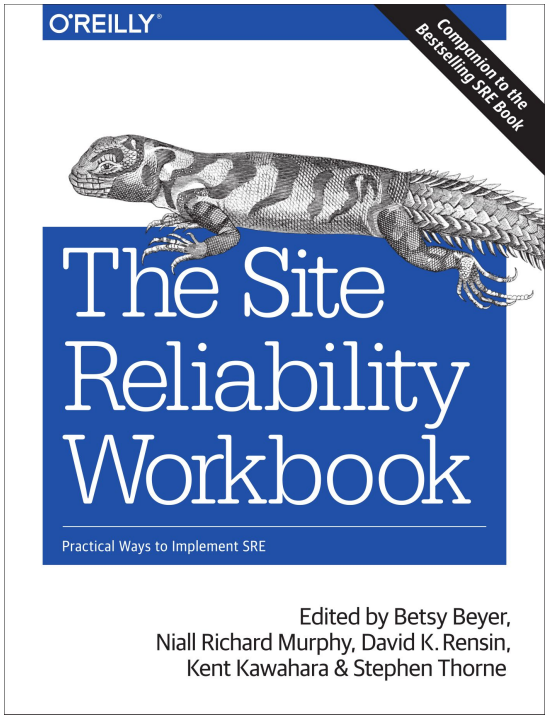
Check out OpenCensus & Stackdriver APM+Trace

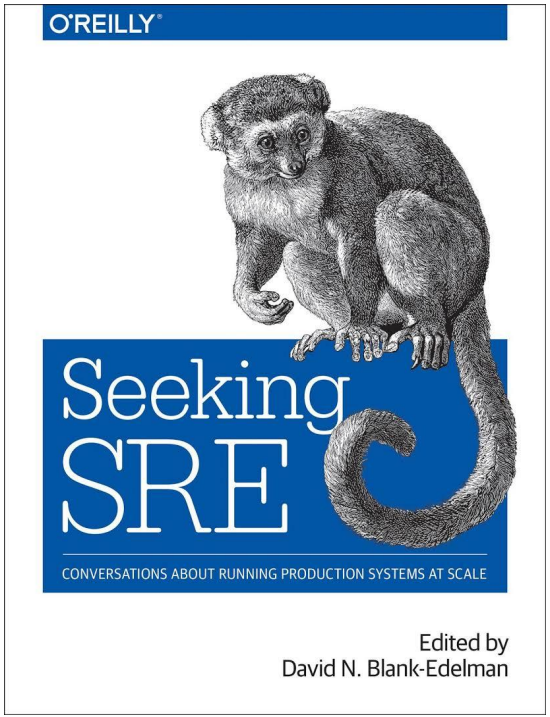
More from the Google SRE team at cloudplatform.googleblog.com

Blog post on observability by Jaana Burcu Dogan ---------->

Google Cloud







Google Cloud

Cover images used with permission. These books can be found on shop.oreilly.com.

Thank you.