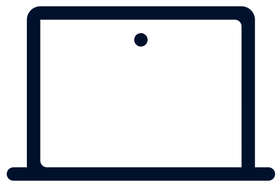
Analyzing Database Performance

Feike Steenbergen, PGConf.EU, Warsaw, 2017-10-25

adyen

wherever people pay

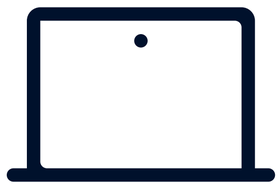

POSTGRESOL
MAERSK
ESTELLE MÆRSK

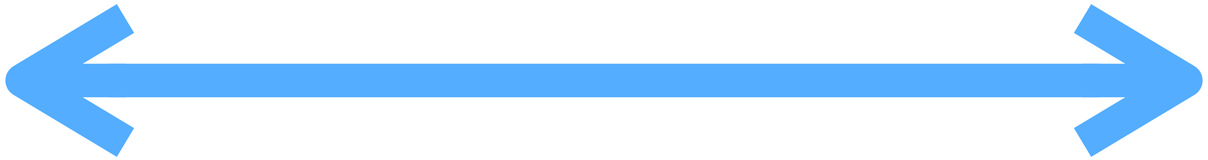


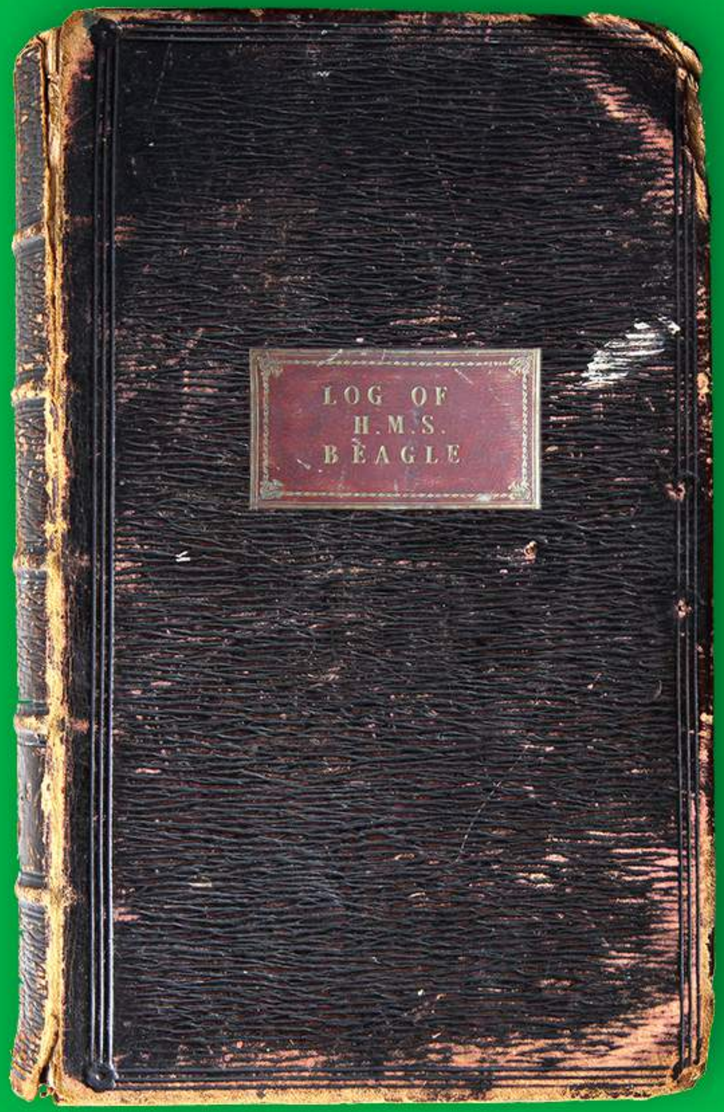
Client

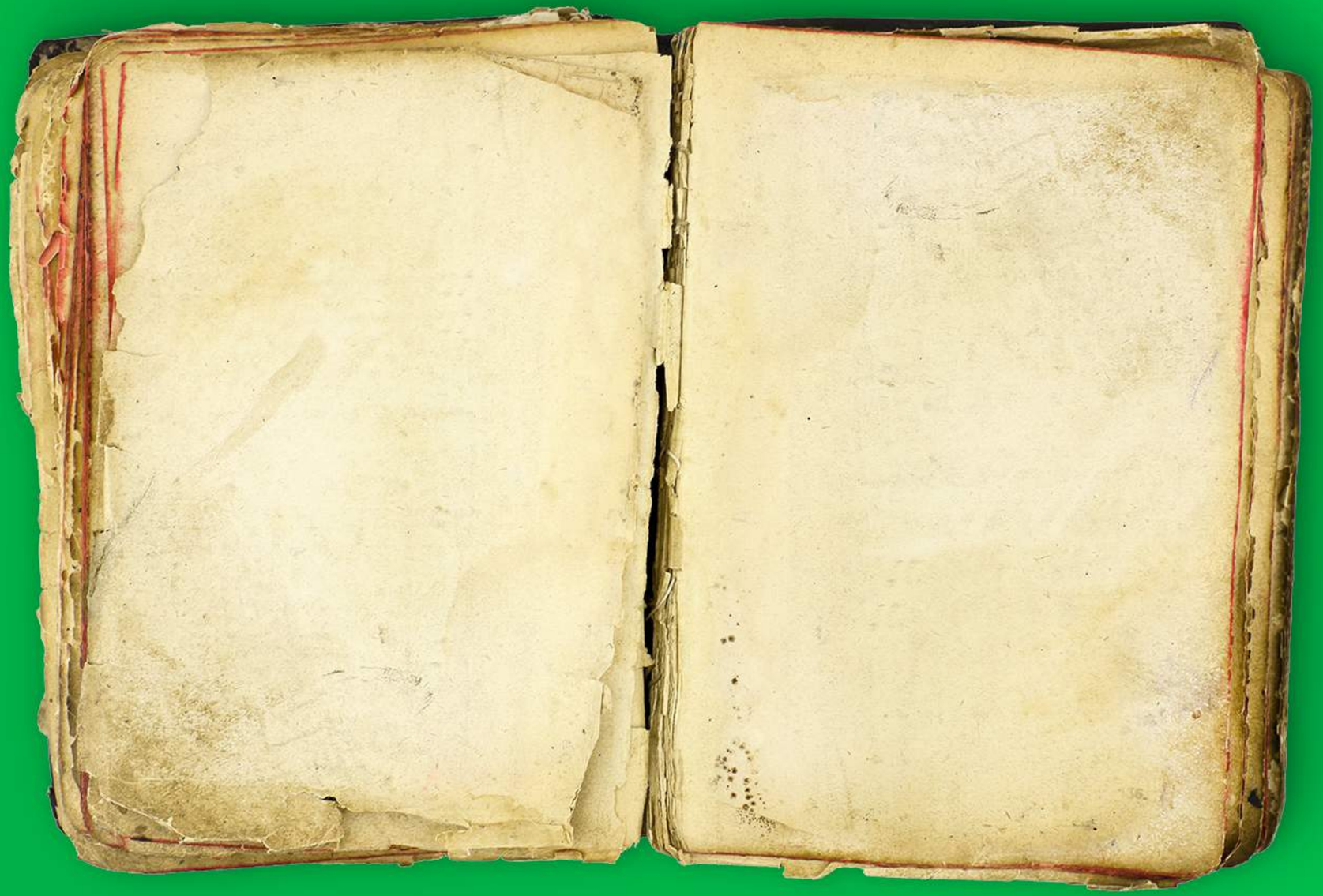
Transport

Database






LOG OF
H.M.S.
BEAGLE



Log settings for performance

```
ALTER SYSTEM SET log_lock_waits TO ‘on';

ALTER SYSTEM SET log_min_duration_statement TO ‘1s';

ALTER SYSTEM SET track_io_timing TO 'on';
```

adyen

Log settings for performance


```
ALTER SYSTEM SET log_lock_waits TO ‘on';

ALTER SYSTEM SET log_min_duration_statement TO ‘1s';

ALTER SYSTEM SET track_io_timing TO 'on';
```

adyen

log entry

```
-[ RECORD 1 ]---+-------------------------------
log_time        | 2017-10-16 09:34:16.037+02
user_name       | feike
connection_from | 192.168.56.1:55271
message         | duration: 2325.732 ms  statement: +
                | SELECT                            +
                |     *                             +
                | FROM                              +
                |     bigtable;
```

When

```
-[ RECORD 1 ]---+-----------------------------------
log_time        | 2017-10-16 09:34:16.037+02
user_name       | feike
connection_from | 192.168.56.1:55271
message         | duration: 2325.732 ms  statement: +
                | SELECT                            +
                |     *                             +
                | FROM                              +
                |     bigtable;
```

Who

```
-[ RECORD 1 ]---+-------------------------------
log_time        | 2017-10-16 09:34:16.037+02
user_name       | feike
connection_from | 192.168.56.1:55271
message         | duration: 2325.732 ms  statement: +
                | SELECT                            +
                |     *                             +
                | FROM                              +
                |     bigtable;
```

Where

```
-[ RECORD 1 ]---+-------------------------------
log_time        | 2017-10-16 09:34:16.037+02
user_name       | feike
connection_from | 192.168.56.1:55271
message         | duration: 2325.732 ms  statement: +
                | SELECT                            +
                |     *                             +
                | FROM                              +
                |     bigtable;
```

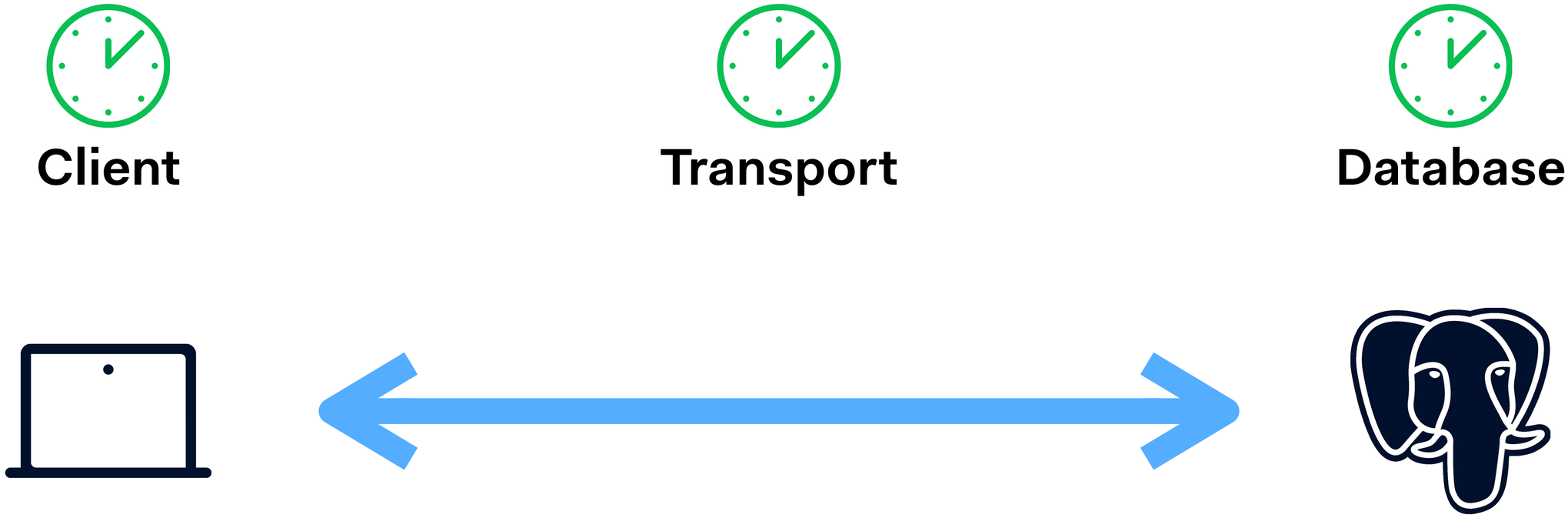
How long

```
-[ RECORD 1 ]---+-----------------------------------
log_time        | 2017-10-16 09:34:16.037+02
user_name       | feike
connection_from | 192.168.56.1:55271
message         | duration: 2325.732 ms  statement: +
                | SELECT                             +
                |     *                              +
                | FROM                               +
                |     bigtable;
```

What

```
-[ RECORD 1 ]---+-------------------------------
log_time        | 2017-10-16 09:34:16.037+02
user_name       | feike
connection_from | 192.168.56.1:55271
message         | duration: 2325.732 ms  statement: +
                | SELECT                             +
                |     *                              +
                | FROM                               +
                |     bigtable;
```

Measuring the whole



Measuring the whole

Client

Transport

Database

psql

```
SELECT
     *
FROM
     bigtable;
```

adyen

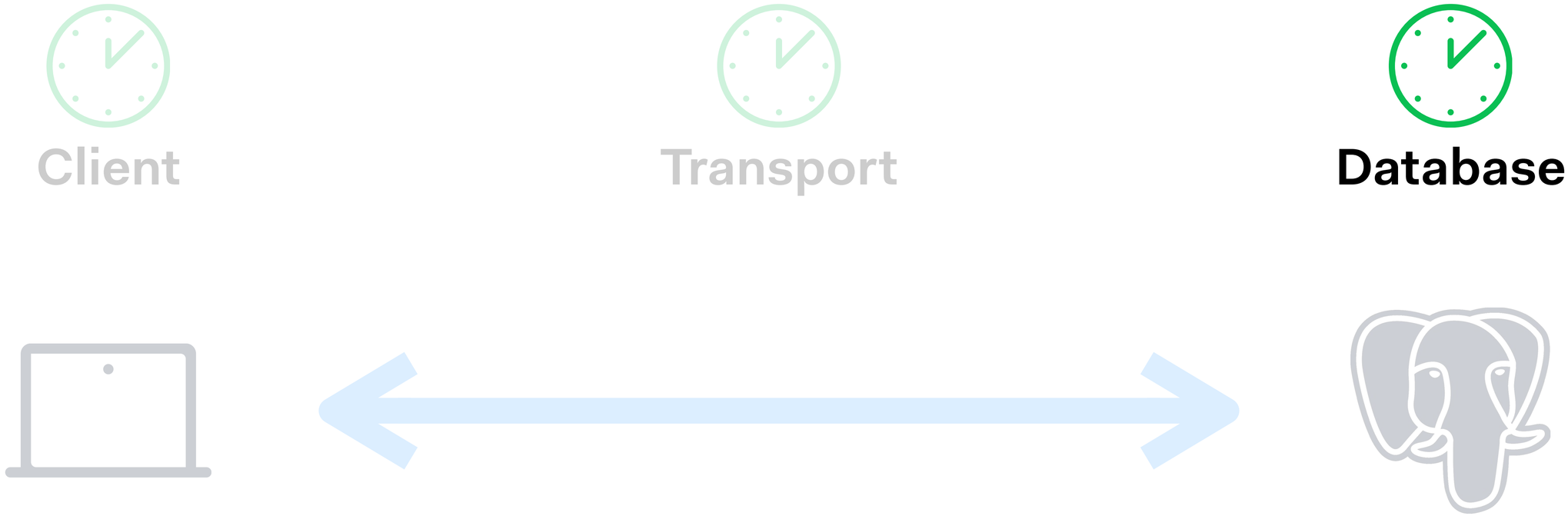
psql

```
SELECT
    *
FROM
    bigtable;

duration: 00:00:11.891
```

adyen

Measuring the database



explain

plan

plan

plan

C

Explain

plan

C

Explain (analyze)

plan

C

Explain

```
EXPLAIN (ANALYZE)
SELECT
    *
FROM
    bigtable;

                                QUERY PLAN
-----------------------------------------------------------------------
 Seq Scan on bigtable
   (cost=0.00..17700.00 rows=1020000 width=32)
   (actual time=191.569..2312.270 rows=30000 loops=1)
 Planning time: 0.068 ms
 Execution time: 2318.109 ms
```


LOST
CONFUSED
UNSURE
UNCLEAR
PERPLEXED
DISORIENTED
BEWILDERED

How

```
Seq Scan on bigtable
   (cost=0.00..17700.00 rows=1020000 width=32)
   (actual time=191.569..2312.270 rows=30000 loops=1)
Planning time: 0.068 ms
Execution time: 2318.109 ms
```

Estimates

```
Seq Scan on bigtable
   (cost=0.00..17700.00 rows=1020000 width=32)
   (actual time=191.569..2312.270 rows=30000 loops=1)
Planning time: 0.068 ms
Execution time: 2318.109 ms
```

Timing

```
Seq Scan on bigtable
   (cost=0.00..17700.00 rows=1020000 width=32)
   (actual time=191.569..2312.270 rows=30000 loops=1)
Planning time: 0.068 ms
Execution time: 2318.109 ms
```

Cost is a range

```
Seq Scan on bigtable
   (cost=0.00..17700.00 rows=1020000 width=32)
   (actual time=191.569..2312.270 rows=30000 loops=1)
Planning time: 0.068 ms
Execution time: 2318.109 ms
```

Time is a range

```
Seq Scan on bigtable
   (cost=0.00..17700.00 rows=1020000 width=32)
   (actual time=191.569..2312.270 rows=30000 loops=1)
Planning time: 0.068 ms
Execution time: 2318.109 ms
```

Reality vs statistics

```
Seq Scan on bigtable
  (cost=0.00..17700.00 rows=1020000 width=32)
  (actual time=191.569..2312.270 rows=30000 loops=1)
Planning time: 0.068 ms
Execution time: 2318.109 ms
```

Total execution time

```
Seq Scan on bigtable
  (cost=0.00..17700.00 rows=1020000 width=32)
  (actual time=191.569..2312.270 rows=30000 loops=1)
Planning time: 0.068 ms
Execution time: 2318.109 ms
```

Less simple query plan

```
Nested Loop
  ->  Nested Loop
    ->  Hash Join
        Hash Cond: (order_item.article_id = article.article_id)
        ->  Seq Scan on order_item
          Filter: ((article_id >= 1521) AND (article_id <= 5432))
        ->  Hash
          ->  Seq Scan on article
              Filter: ((price % '1'::numeric) = 0.99)
    ->  Index Scan using order_pkey on "order"
        Index Cond: (order_id = order_item.order_id)
  ->  Index Only Scan using customer_pkey on customer
    Index Cond: (customer_id = "order".customer_id)
```

Sequential Scan

```
Nested Loop
  ->  Nested Loop
    ->  Hash Join
        Hash Cond: (order_item.article_id = article.article_id)
        ->  Seq Scan on order_item
            Filter: ((article_id >= 1521) AND (article_id <= 5432))
        ->  Hash
            ->  Seq Scan on article
                Filter: ((price % '1'::numeric) = 0.99)
    ->  Index Scan using order_pkey on "order"
        Index Cond: (order_id = order_item.order_id)
  ->  Index Only Scan using customer_pkey on customer
    Index Cond: (customer_id = "order".customer_id)
```

adyen

Hash of the Sequential Scan

```
Nested Loop
  ->  Nested Loop
    ->  Hash Join
        Hash Cond: (order_item.article_id = article.article_id)
        ->  Seq Scan on order_item
          Filter: ((article_id >= 1521) AND (article_id <= 5432))
        ->  Hash
          ->  Seq Scan on article
              Filter: ((price % '1'::numeric) = 0.99)
    ->  Index Scan using order_pkey on "order"
        Index Cond: (order_id = order_item.order_id)
  ->  Index Only Scan using customer_pkey on customer
    Index Cond: (customer_id = "order".customer_id)
```

Sequantial scan

```
Nested Loop
  ->  Nested Loop
    ->  Hash Join
        Hash Cond: (order_item.article_id = article.article_id)
        ->  Seq Scan on order_item
          Filter: ((article_id >= 1521) AND (article_id <= 5432))
        ->  Hash
          ->  Seq Scan on article
              Filter: ((price % '1'::numeric) = 0.99)
    ->  Index Scan using order_pkey on "order"
        Index Cond: (order_id = order_item.order_id)
  ->  Index Only Scan using customer_pkey on customer
    Index Cond: (customer_id = "order".customer_id)
```

Hash Join

```
Nested Loop
  ->  Nested Loop
    ->  Hash Join
        Hash Cond: (order_item.article_id = article.article_id)
        ->  Seq Scan on order_item
              Filter: ((article_id >= 1521) AND (article_id <= 5432))
        ->  Hash
              ->  Seq Scan on article
                    Filter: ((price % '1'::numeric) = 0.99)
    ->  Index Scan using order_pkey on "order"
        Index Cond: (order_id = order_item.order_id)
  ->  Index Only Scan using customer_pkey on customer
    Index Cond: (customer_id = "order".customer_id)
```

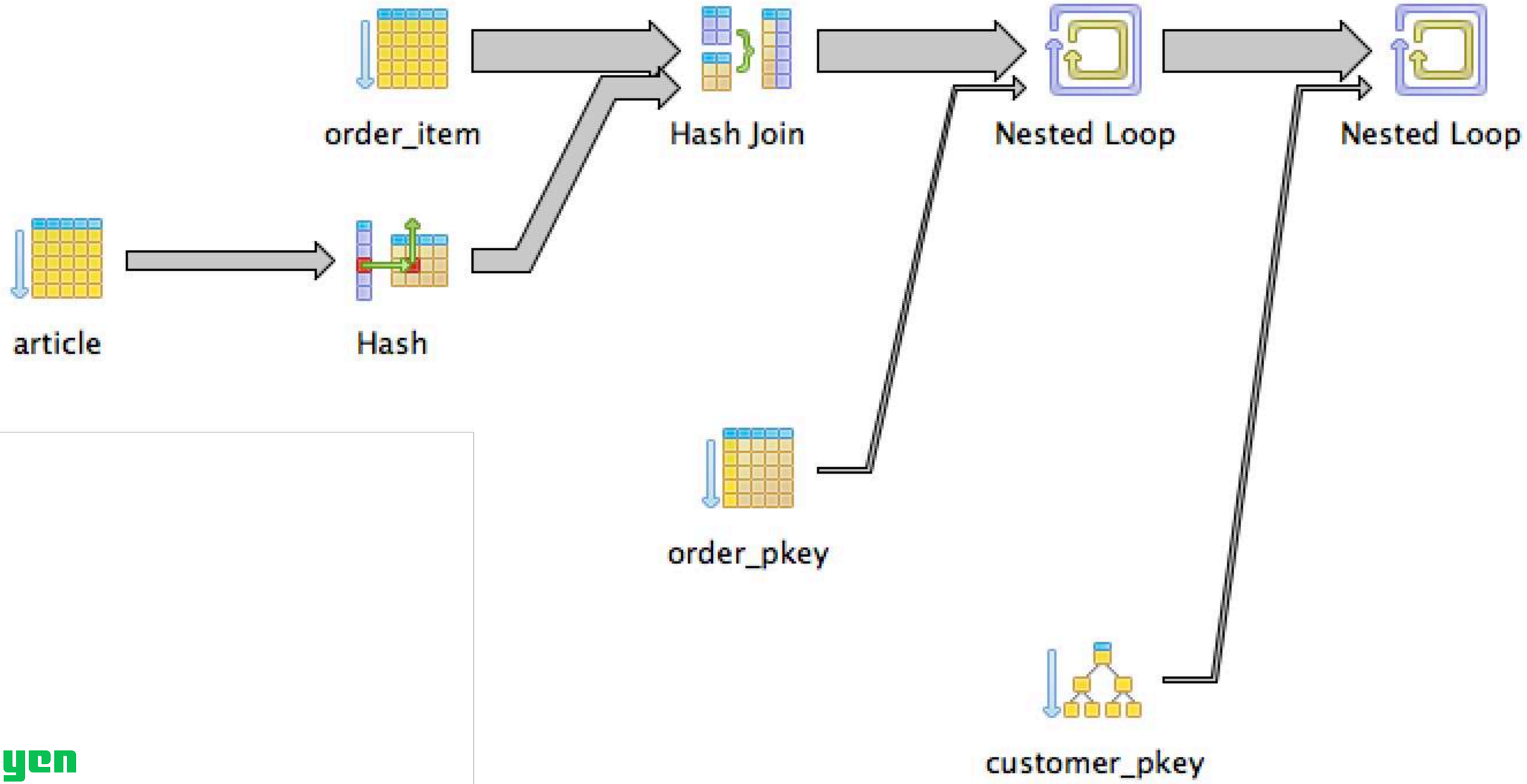
Nested Loop

```
Nested Loop
  ->  Nested Loop
    ->  Hash Join
        Hash Cond: (order_item.article_id = article.article_id)
        ->  Seq Scan on order_item
          Filter: ((article_id >= 1521) AND (article_id <= 5432))
        ->  Hash
          ->  Seq Scan on article
              Filter: ((price % '1'::numeric) = 0.99)
    ->  Index Scan using order_pkey on "order"
        Index Cond: (order_id = order_item.order_id)
  ->  Index Only Scan using customer_pkey on customer
    Index Cond: (customer_id = "order".customer_id)
```

Nested Loop

```
Nested Loop
  ->  Nested Loop
    ->  Hash Join
        Hash Cond: (order_item.article_id = article.article_id)
        ->  Seq Scan on order_item
          Filter: ((article_id >= 1521) AND (article_id <= 5432))
        ->  Hash
          ->  Seq Scan on article
              Filter: ((price % '1'::numeric) = 0.99)
    ->  Index Scan using order_pkey on "order"
        Index Cond: (order_id = order_item.order_id)
  ->  Index Only Scan using customer_pkey on customer
    Index Cond: (customer_id = "order".customer_id)
```

pgAdmin



adyen

explain.depesz.com

Result: p89 : Demo Explain Plan

Settings

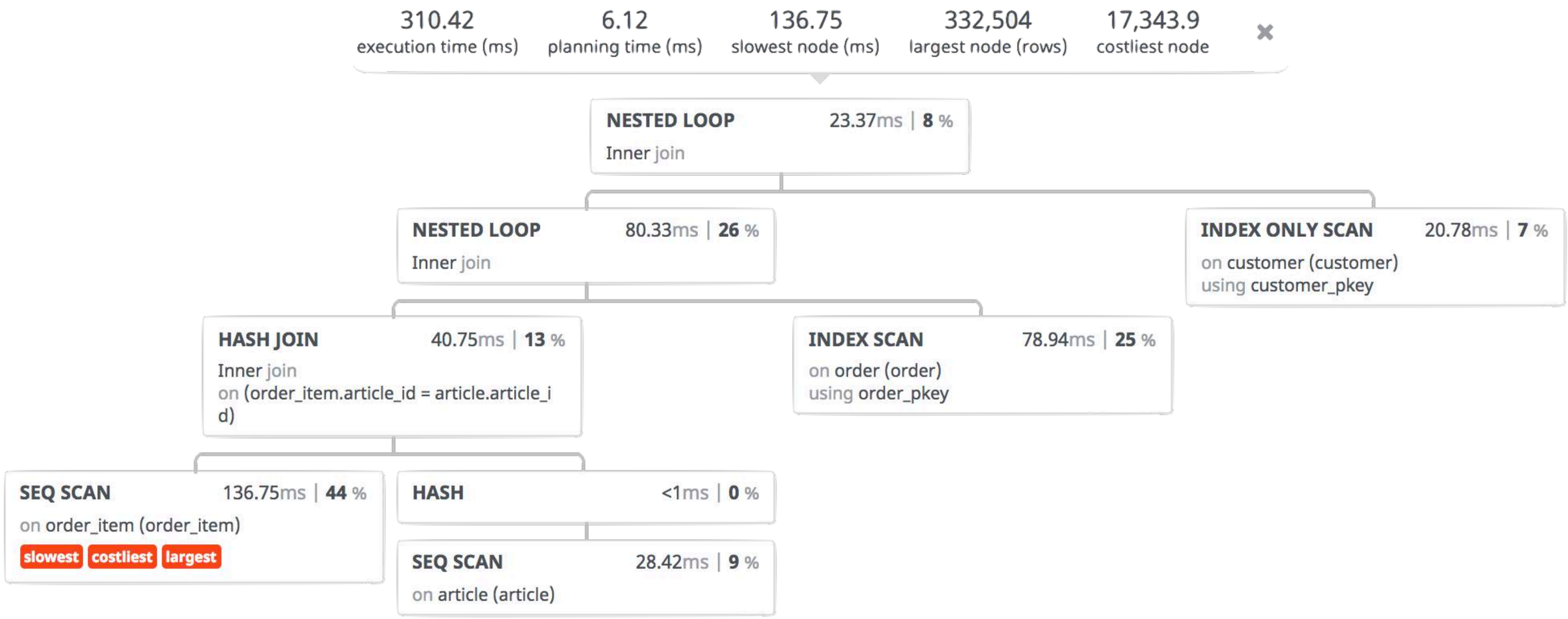
HTML | TEXT | STATS

Add optimization

#	exclusive	inclusive	rows x	rows	loops	node
1.	3.065	175.672	↓ 2.5	4,109	1	→ Nested Loop (cost=206.34..20,170.45 rows=1,647 width=30) (actual time=4.450..175.672 rows=4,109 loops=1)
2.	0.000	164.389	↓ 2.5	4,109	1	→ Nested Loop (cost=206.05..19,633.94 rows=1,647 width=30) (actual time=4.442..164.389 rows=4,109 loops=1)
3.	29.210	148.260	↓ 2.5	4,109	1	→ Hash Join (cost=205.62..18,801.62 rows=1,647 width=18) (actual time=4.433..148.260 rows=4,109 loops=1) Hash Cond: (order_item.article_id = article.article_id)
4.	114.681	114.681	↓ 1.0	332,383	1	→ Seq Scan on order_item (cost=0.00..17,344.15 rows=329,434 width=12) (actual time=0.014..114.681 rows=332,383 loops=1) Filter: ((article_id >= 1521) AND (article_id <= 5432)) Rows Removed by Filter: 517560
5.	0.022	4.369	↓ 2.4	120	1	→ Hash (cost=205.00..205.00 rows=50 width=10) (actual time=4.369..4.369 rows=120 loops=1) Buckets: 1024 Batches: 1 Memory Usage: 14kB
6.	4.347	4.347	↓ 2.4	120	1	→ Seq Scan on article (cost=0.00..205.00 rows=50 width=10) (actual time=0.017..4.347 rows=120 loops=1) Filter: ((price % '1'::numeric) = 0.99) Rows Removed by Filter: 9880
7.	16.436	16.436	↑ 1.0	1	4,109	→ Index Scan using order_pkey on "order" (cost=0.42..0.50 rows=1 width=16) (actual time=0.003..0.004 rows=1 loops=4,109) Index Cond: (order_id = order_item.order_id)
8.	8.218	8.218	↑ 1.0	1	4,109	→ Index Only Scan using customer_pkey on customer (cost=0.29..0.32 rows=1 width=4) (actual time=0.002..0.002 rows=1 loops=4,109) Index Cond: (customer_id = "order".customer_id) Heap Fetches: 4109

adyen

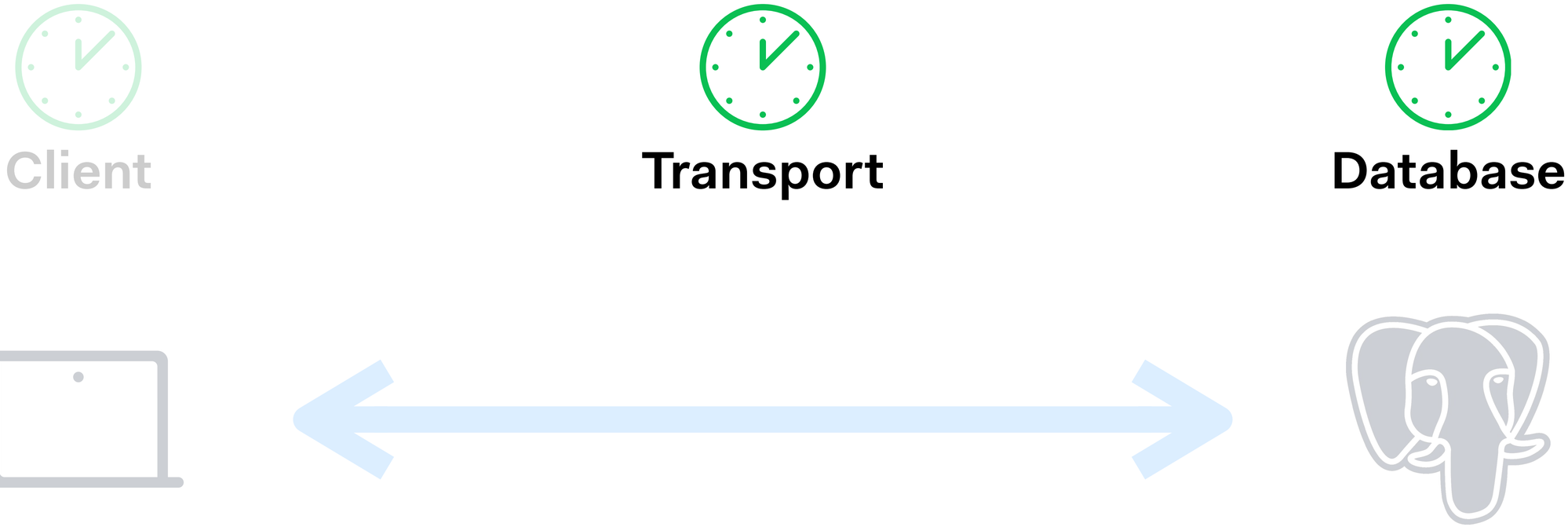
pev



http:///tatiyants.com/pev

adyen

Measure without a client



Python as a client

```
 1 import psycopg2
 2 import time
 3
 4 connection = psycopg2.connect(dbname='f1db')
 5 cursor = connection.cursor(name='mycursor')
 6
 7 started = time.time()
 8 cursor.execute('SELECT * FROM bigtable')
 9 for row in cursor:
10     pass
11 finished = time.time()
12
13 print('duration: {0}'.format(finished - started))
```

Measure

```
$ python dummy_client.py
duration: 2.5234987
```

Database

2.3 seconds

adyen

Client

9.4 seconds

2.3 seconds

adyen

Transport

9.4 seconds

0.2 seconds

2.3 seconds

adyen

Time spent

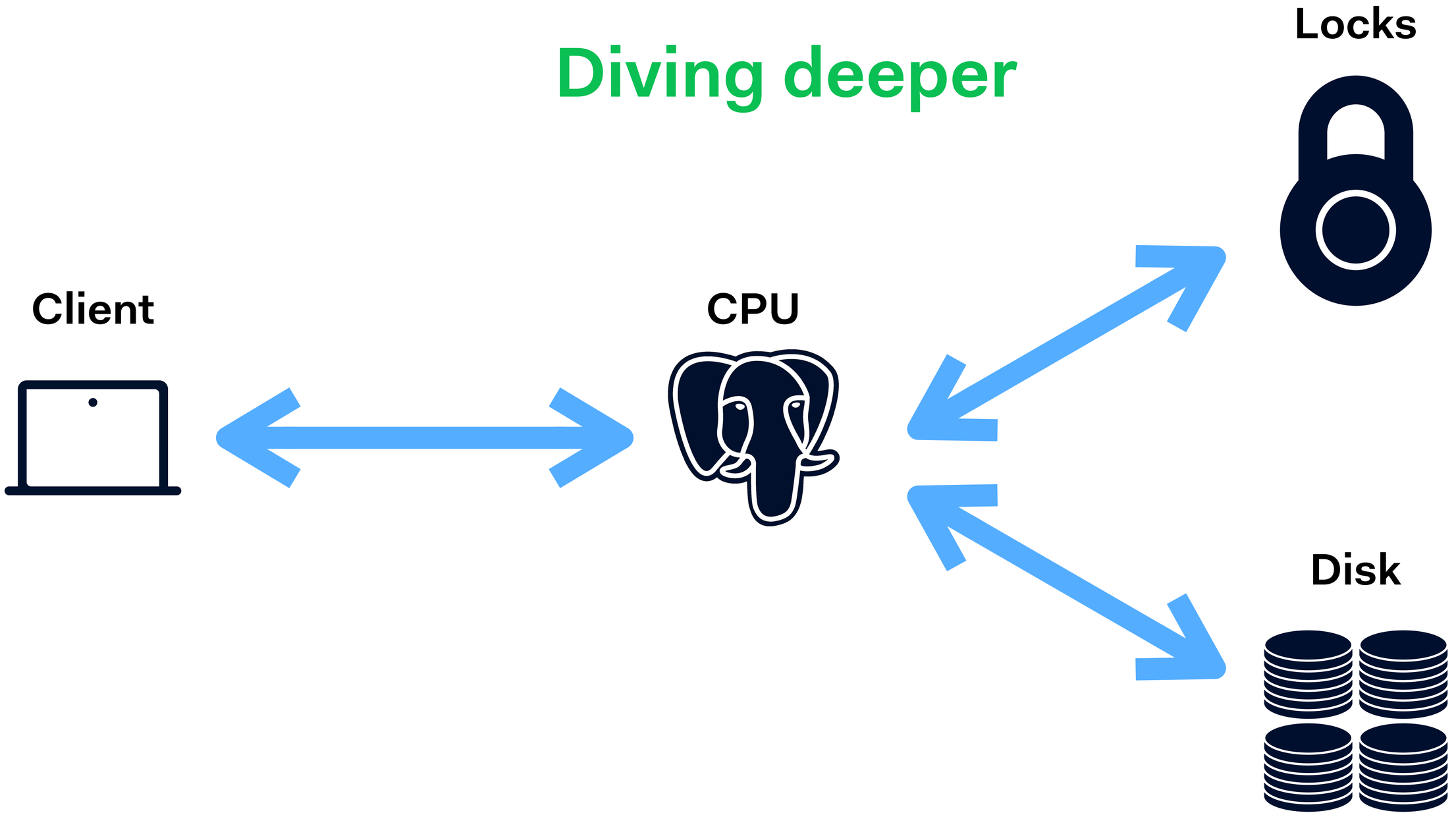
79 %

2 %

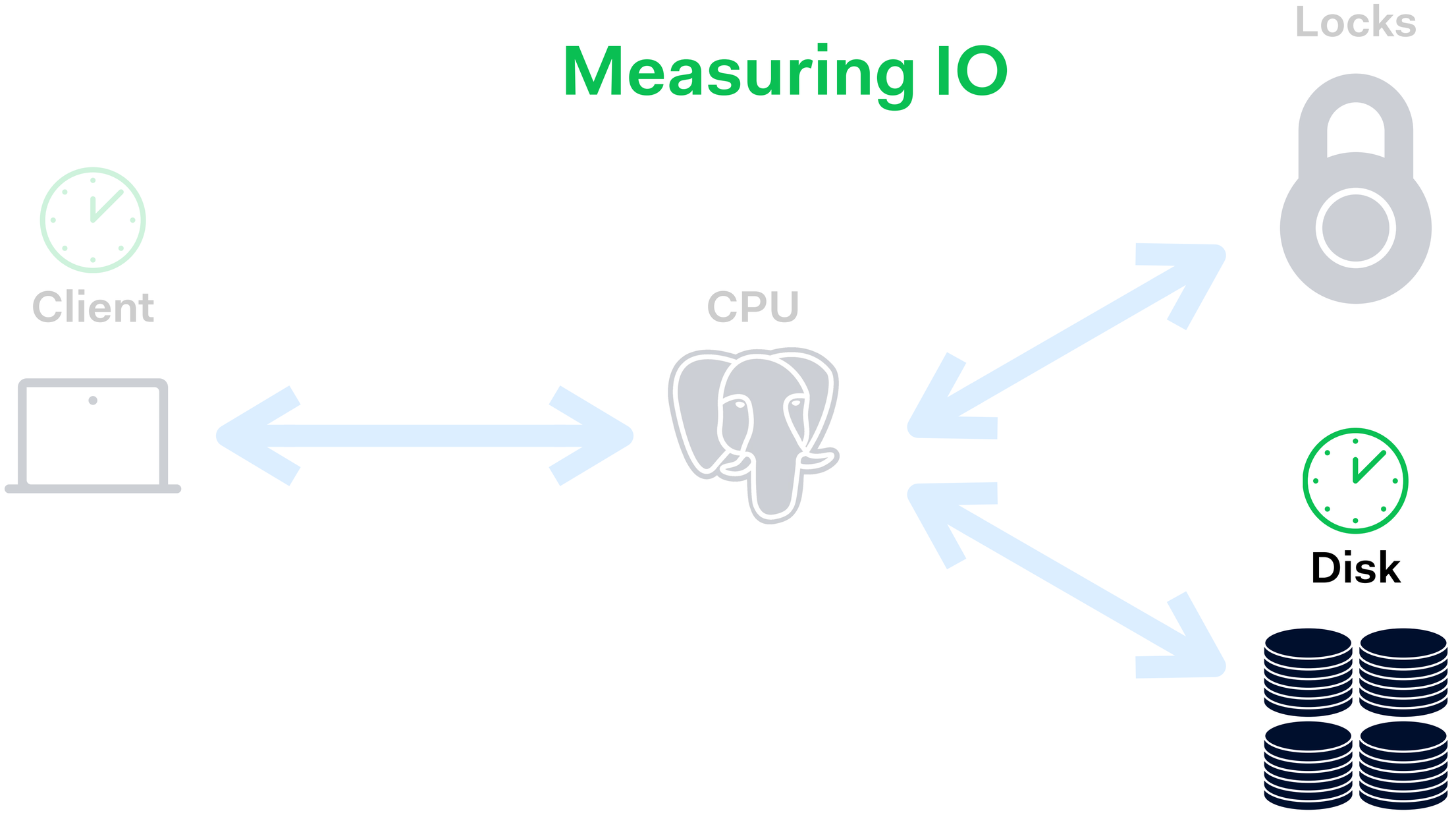
19 %

adyen

Diving deeper



Measuring IO



Explain with Buffers

```
EXPLAIN (ANALYZE, BUFFERS, TIMING OFF, COSTS OFF)
SELECT
    *
FROM
    laptimes;
```

adyen

Buffers hit / read

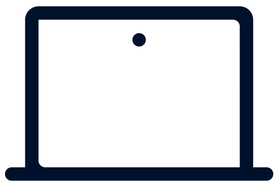
```
EXPLAIN (ANALYZE, BUFFERS, TIMING OFF, COSTS OFF)
SELECT
     *
FROM
     laptimes;
```

Query plan

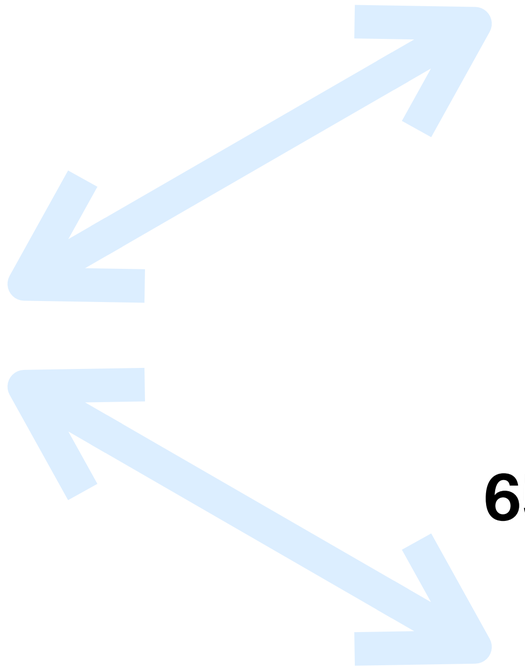
```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared read=2690
   I/O Timings: read=652.826
 Planning time: 3.940 ms
 Execution time: 722.450 ms
```

Time spent

722 ms (100%)

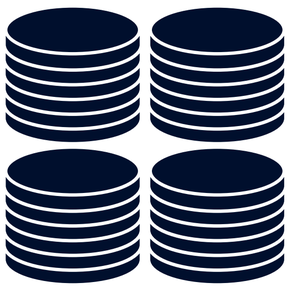


CPU



Locks

653 ms (90%)



adyen

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared read=2690
   I/O Timings: read=652.826
 Planning time: 3.940 ms
 Execution time: 722.450 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=32 read=2658
   I/O Timings: read=731.289
 Planning time: 0.045 ms
 Execution time: 775.574 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=64 read=2626
   I/O Timings: read=808.450
 Planning time: 0.051 ms
 Execution time: 869.010 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=96 read=2594
   I/O Timings: read=650.151
 Planning time: 0.055 ms
 Execution time: 703.729 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=128 read=2562
   I/O Timings: read=656.976
 Planning time: 0.043 ms
 Execution time: 698.512 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=160 read=2530
   I/O Timings: read=843.544
 Planning time: 0.044 ms
 Execution time: 896.112 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=192 read=2498
   I/O Timings: read=656.763
 Planning time: 0.072 ms
 Execution time: 701.574 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=256 read=2434
   I/O Timings: read=738.955
 Planning time: 0.077 ms
 Execution time: 782.450 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=288 read=2402
   I/O Timings: read=760.545
 Planning time: 0.043 ms
 Execution time: 802.931 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=320 read=2370
   I/O Timings: read=724.063
 Planning time: 0.049 ms
 Execution time: 770.126 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=352 read=2338
   I/O Timings: read=752.979
 Planning time: 0.046 ms
 Execution time: 816.250 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=384 read=2306
   I/O Timings: read=673.584
 Planning time: 0.049 ms
 Execution time: 723.396 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=416 read=2274
   I/O Timings: read=822.325
 Planning time: 0.052 ms
 Execution time: 866.023 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=448 read=2242
   I/O Timings: read=513.394
 Planning time: 0.052 ms
 Execution time: 551.115 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=480 read=2210
   I/O Timings: read=869.133
 Planning time: 0.044 ms
 Execution time: 907.231 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=512 read=2178
   I/O Timings: read=639.766
 Planning time: 0.043 ms
 Execution time: 686.265 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=544 read=2146
   I/O Timings: read=652.697
 Planning time: 0.051 ms
 Execution time: 698.870 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=576 read=2114
   I/O Timings: read=655.188
 Planning time: 0.046 ms
 Execution time: 703.512 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=608 read=2082
   I/O Timings: read=1100.775
 Planning time: 0.046 ms
 Execution time: 1144.916 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=640 read=2050
   I/O Timings: read=884.720
 Planning time: 0.048 ms
 Execution time: 929.228 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=768 read=1922
   I/O Timings: read=663.799
 Planning time: 0.041 ms
 Execution time: 702.821 ms
```

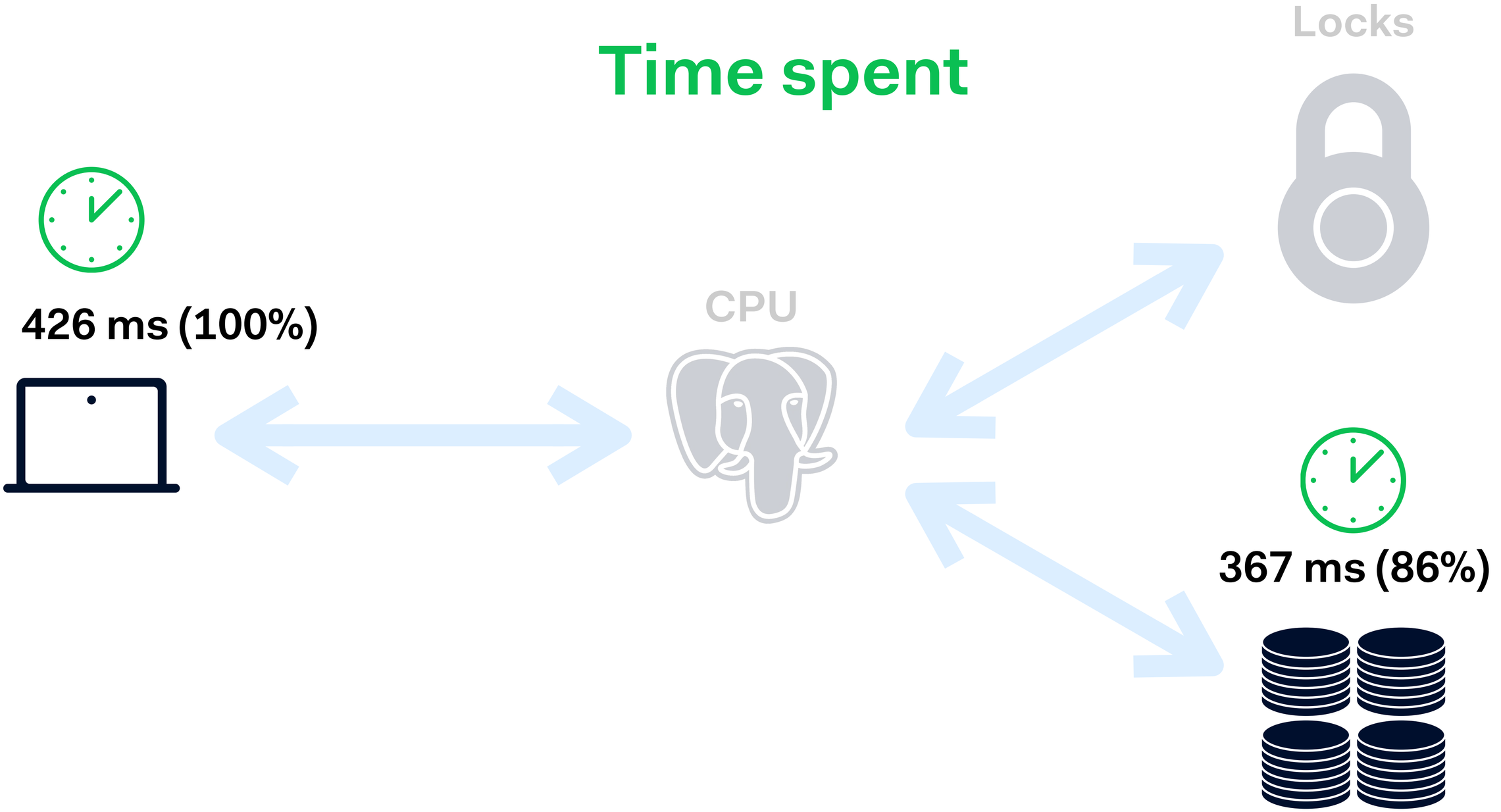
Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=864 read=1826
   I/O Timings: read=840.873
 Planning time: 0.043 ms
 Execution time: 884.622 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=960 read=1730
   I/O Timings: read=367.398
 Planning time: 0.073 ms
 Execution time: 426.409 ms
```

Time spent



Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=1120 read=1570
   I/O Timings: read=455.304
 Planning time: 0.042 ms
 Execution time: 490.674 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=1280 read=1410
   I/O Timings: read=644.752
 Planning time: 0.044 ms
 Execution time: 767.250 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=1504 read=1186
   I/O Timings: read=549.714
 Planning time: 0.051 ms
 Execution time: 589.280 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=1728 read=962
   I/O Timings: read=354.424
 Planning time: 0.045 ms
 Execution time: 399.210 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=2073 read=617
   I/O Timings: read=1.812
 Planning time: 0.067 ms
 Execution time: 80.141 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=2393 read=297
   I/O Timings: read=0.594
 Planning time: 0.049 ms
 Execution time: 55.776 ms
```

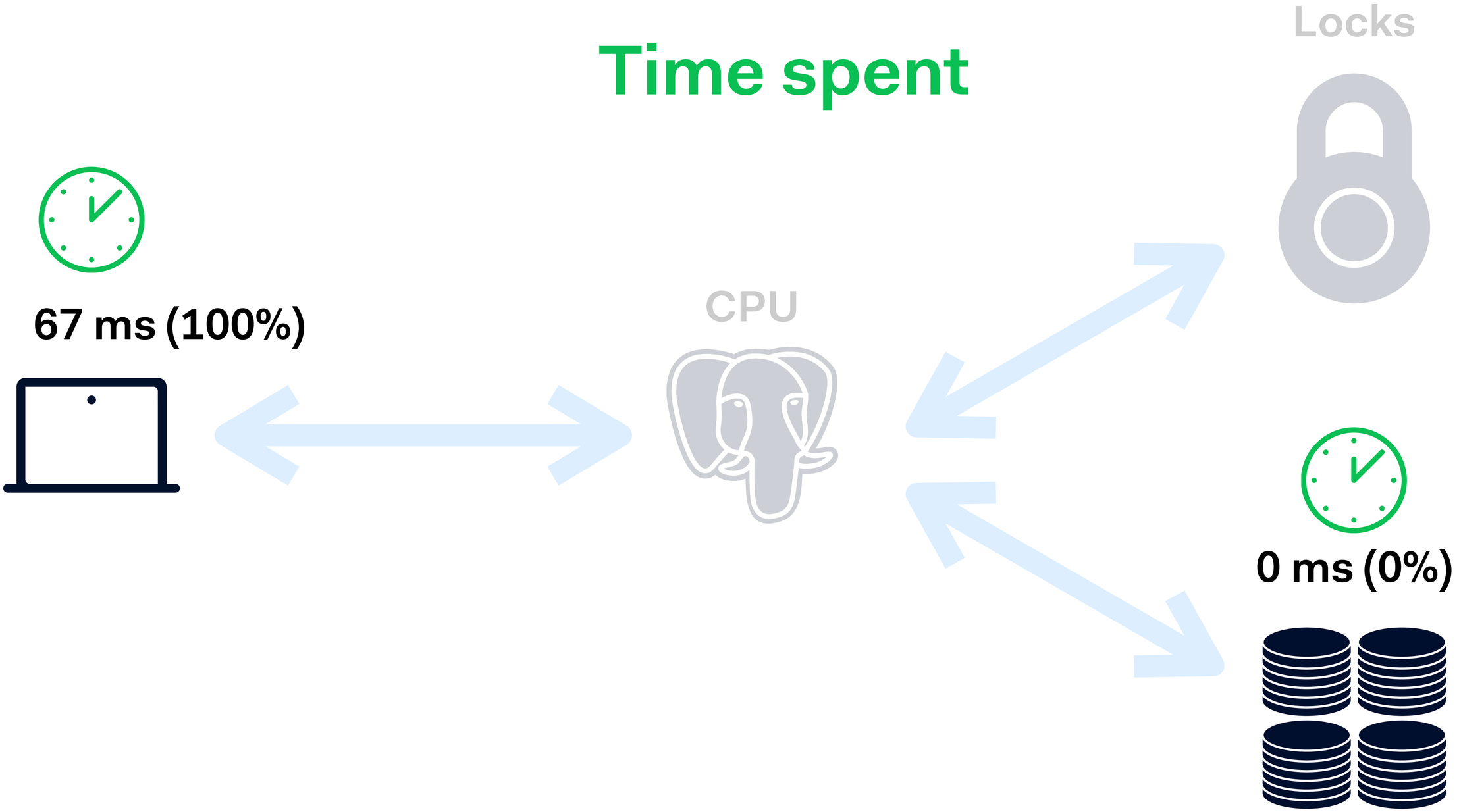
Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=2681 read=9
   I/O Timings: read=0.049
 Planning time: 0.045 ms
 Execution time: 40.774 ms
```

Query plan

```
Seq Scan on laptimes (actual rows=422277 loops=1)
   Buffers: shared hit=2690
 Planning time: 0.348 ms
 Execution time: 67.245 ms
```

Time spent

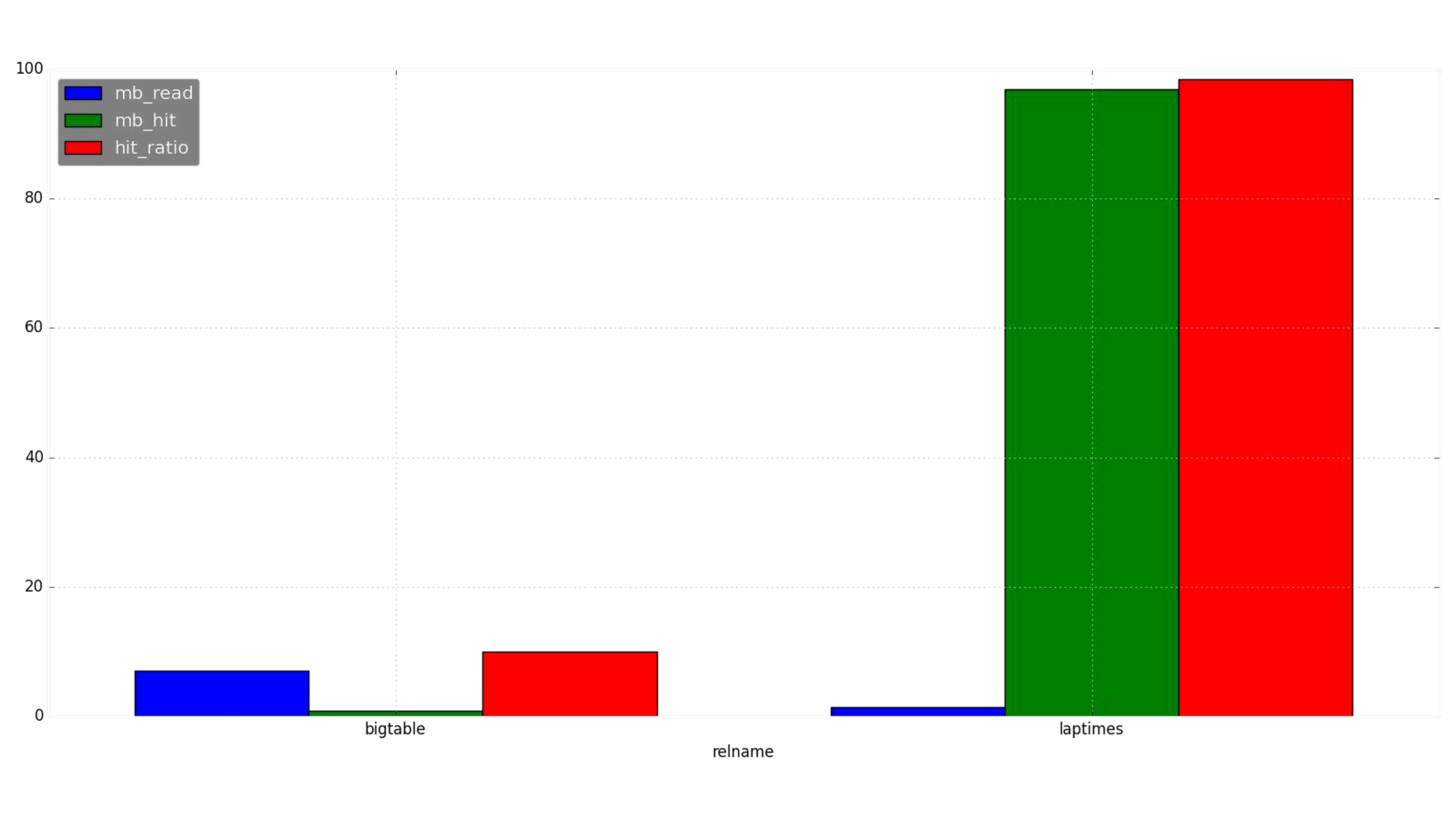



DB overview
Zoom Out
Last 1 hour
dbname
test
TPS
Last 6 hours
0.9
QPS
Last 6 hours
0.57
Query runtime
Last 6 hours
0 ms
DB size ch. 1h
Last 6 hours
0 B
Approx Bloat
Last 6 hours
692.3 KiB
CPU Load
Last 6 hours
1.6%
Tuple IUD statistics + IO times (1h ratios)
DELETE Avg: 0
UPDATE Avg: 0
INSERT Avg: 0
blk_write_time Avg: 0 ms
blk_read_time Avg: 137 ms
Buffer hit ratio + Rollback ratio
Shared buffers hit ratio Avg: 94.2%
TX rollback ratio Avg: 0%
Backends + Deadlocks + Temp bytes
Deadlocks (1h rate)
#Backends
Temp bytes written
WAL rate + DB size
WAL rate Avg: 0 Bps
DB Size Avg: 100.4 MiB
Sessions
idle Avg: 2.0
idle in transaction Avg: 0
waiting Avg: 0
active Avg: 1.0
CPU load + avg. query runtime
load_5 Avg: 0.96%
avg_query_runtime Avg: 0 ms

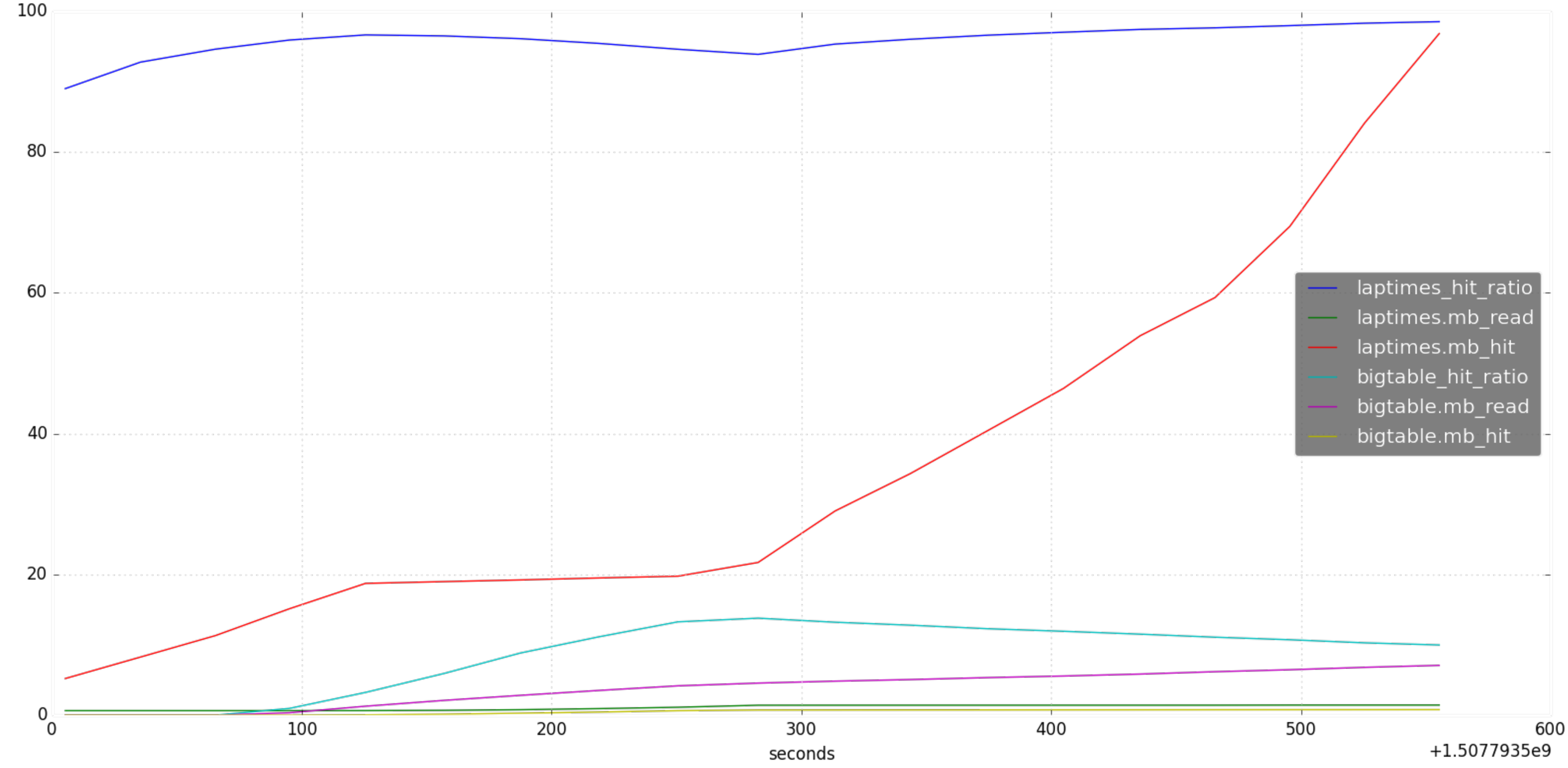
Blu

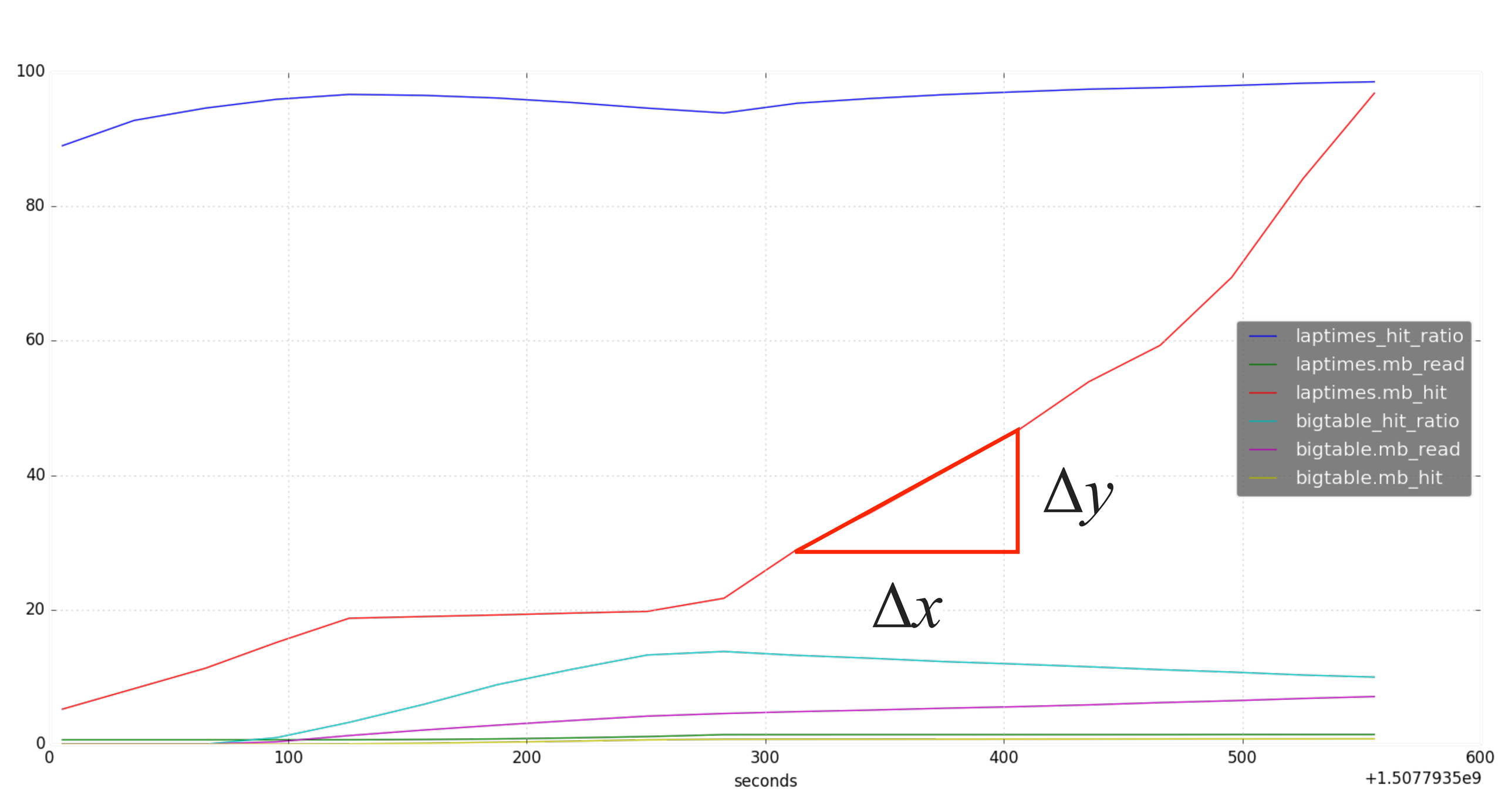

HARDING'S IMPROVED COUNTER
PATENTED.
8326913


8
100
120
80
140
60
160
40
180
km/h
20
200
0
220
ODO
10889

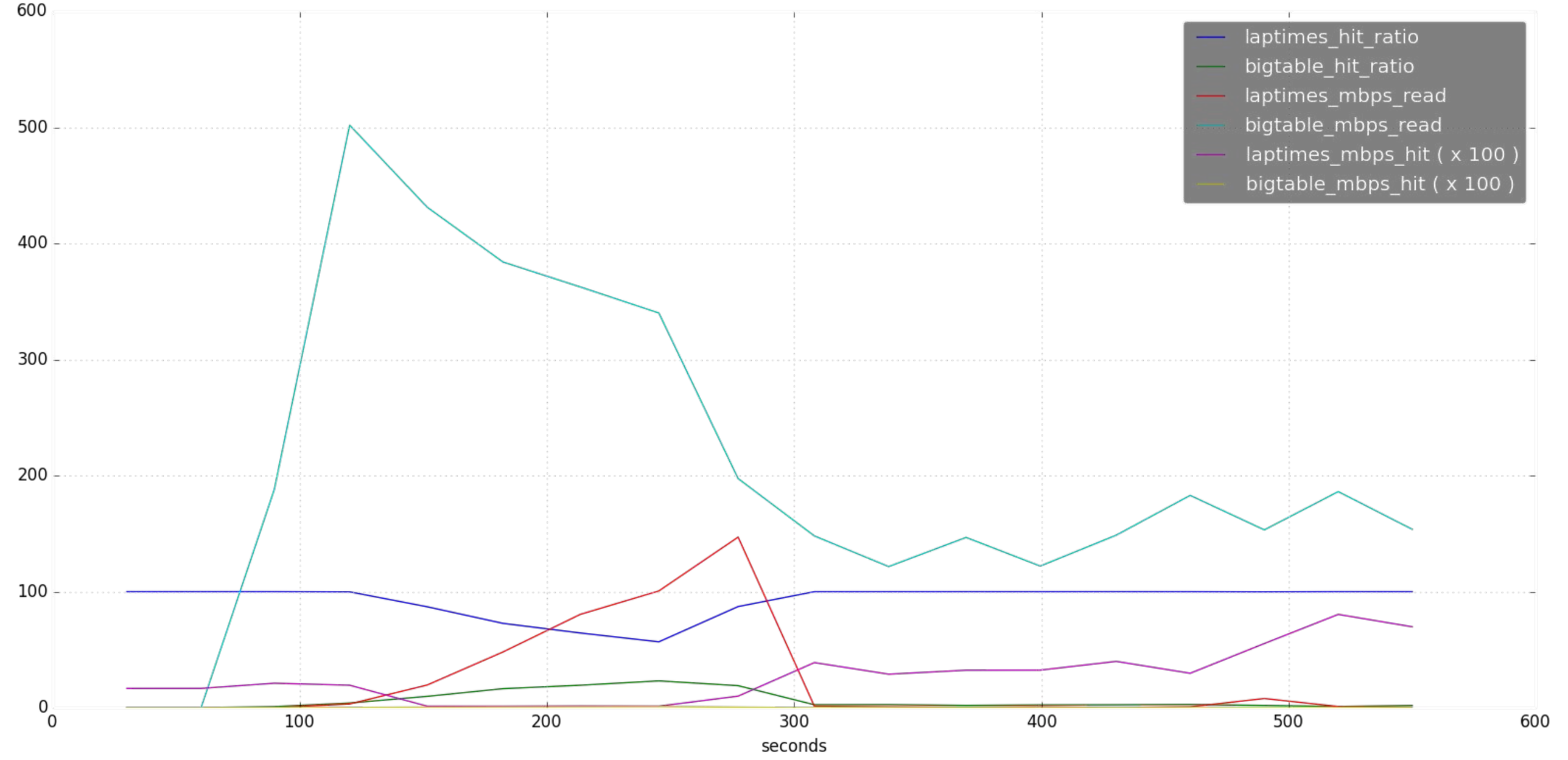

mb_read
mb_hit
hit_ratio
100
80
60
40
20
0
bigtable
laptimes
relname


Canon
Nikon
Nikon


100
80
60
40
20
0
0
100
200
300
400
500
600
seconds
+1.5077935e9
laptimes_hit_ratio
laptimes.mb_read
laptimes.mb_hit
bigtable_hit_ratio
bigtable.mb_read
bigtable.mb_hit


100
80
60
40
20
0
0
100
200
300
400
500
600
seconds
+1.5077935e9
laptimes_hit_ratio
laptimes.mb_read
laptimes.mb_hit
bigtable_hit_ratio
bigtable.mb_read
bigtable.mb_hit
Δy
Δx


8
100
120
80
140
60
160
km/h
40
180
20
200
0
ODO
10889

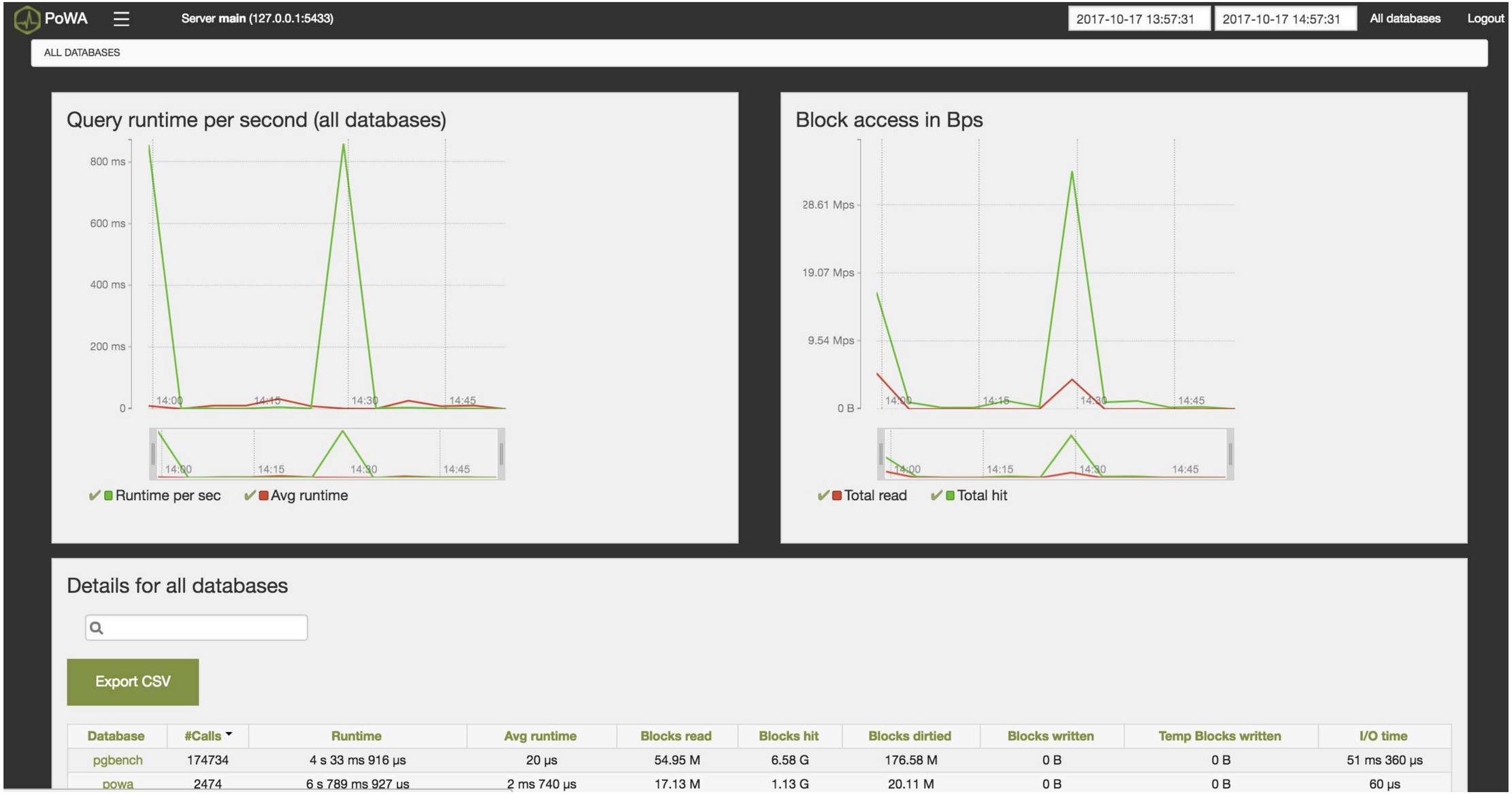

laptimes_hit_ratio
bigtable_hit_ratio
laptimes_mbps_read
bigtable_mbps_read
laptimes_mbps_hit (x 100)
bigtable_mbps_hit (x 100)
600
500
400
300
200
100
0
0
100
200
300
400
500
600
seconds

pgwatch2

https://github.com/cybertec-postgresql/pgwatch2

adyen

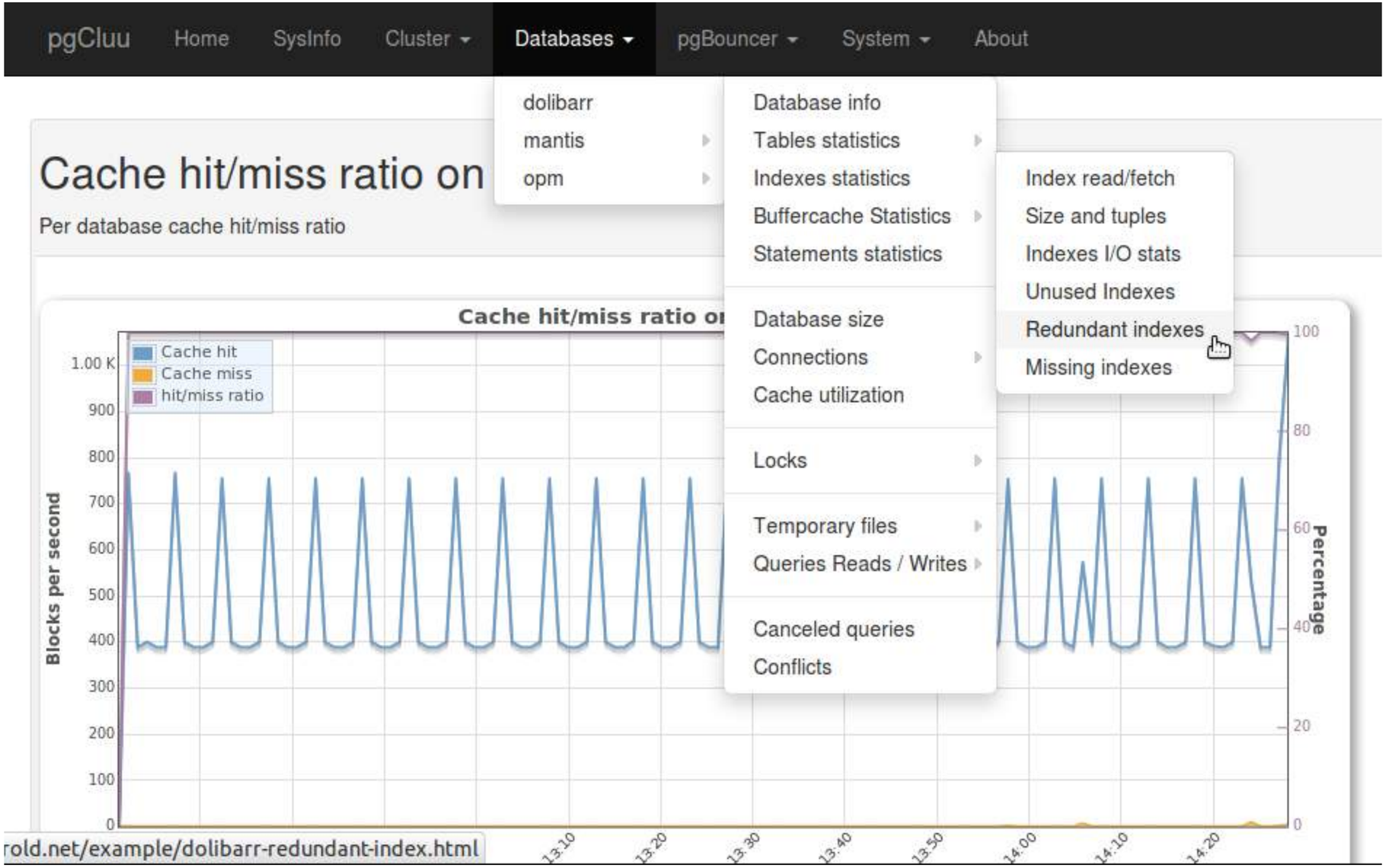
PostgreSQL Workload Analyzer (POWA)



https://github.com/dalibo/powa

adyen

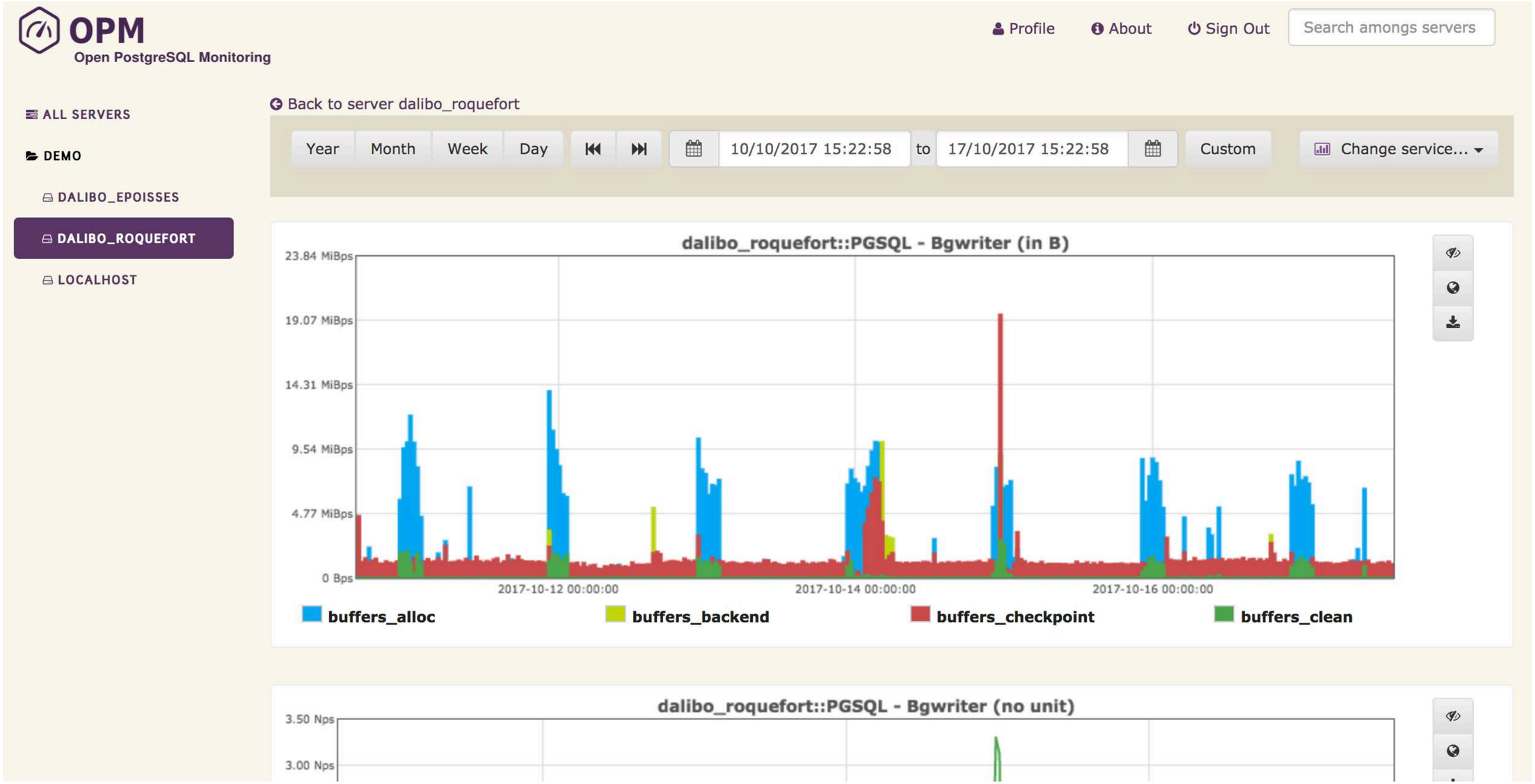
pgCluu



https://github.com/darold/pgcluu/

adyen

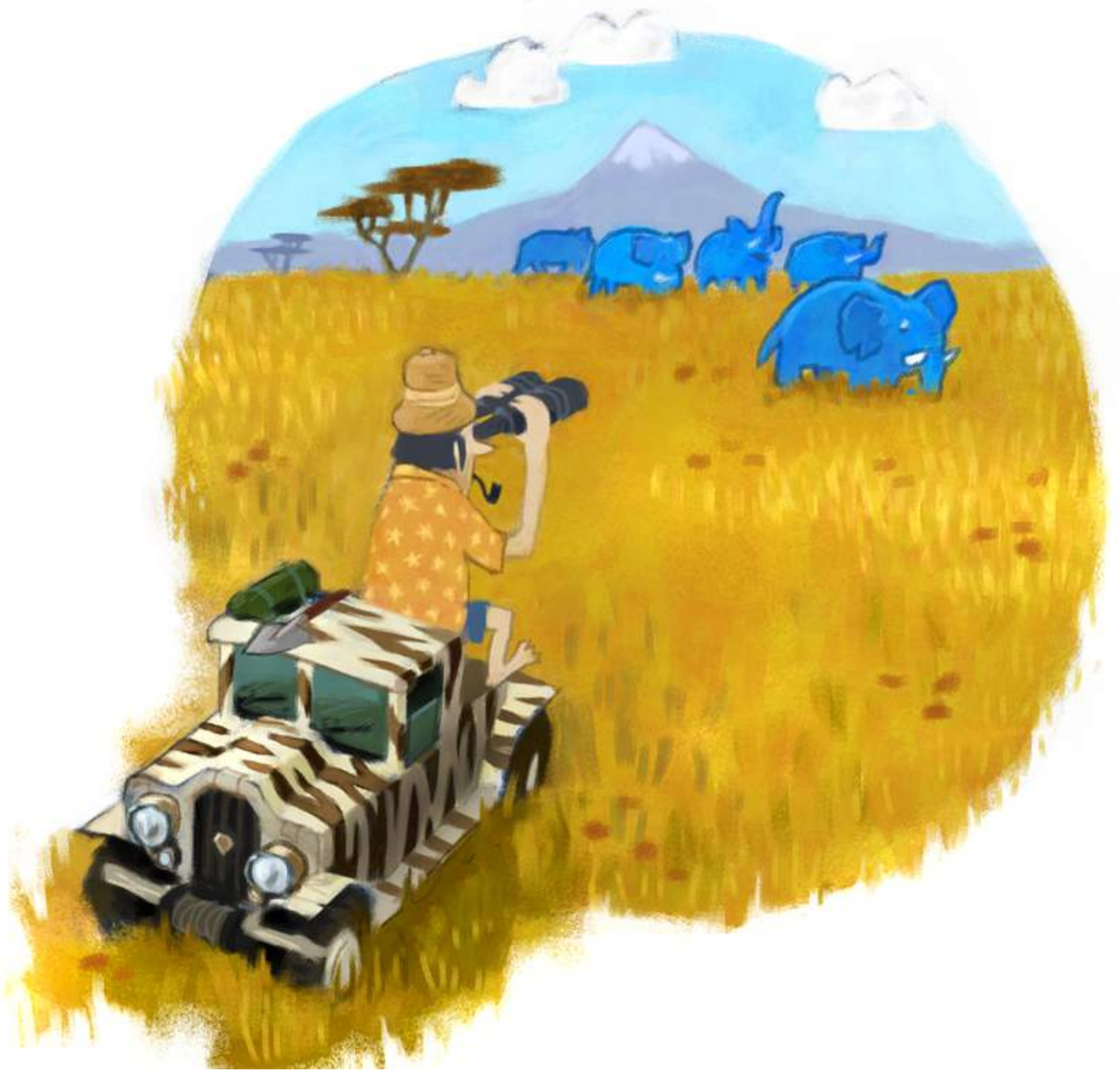
Open PostgreSQL Monitoring (OPM)



https://opm.readthedocs.io/

adyen

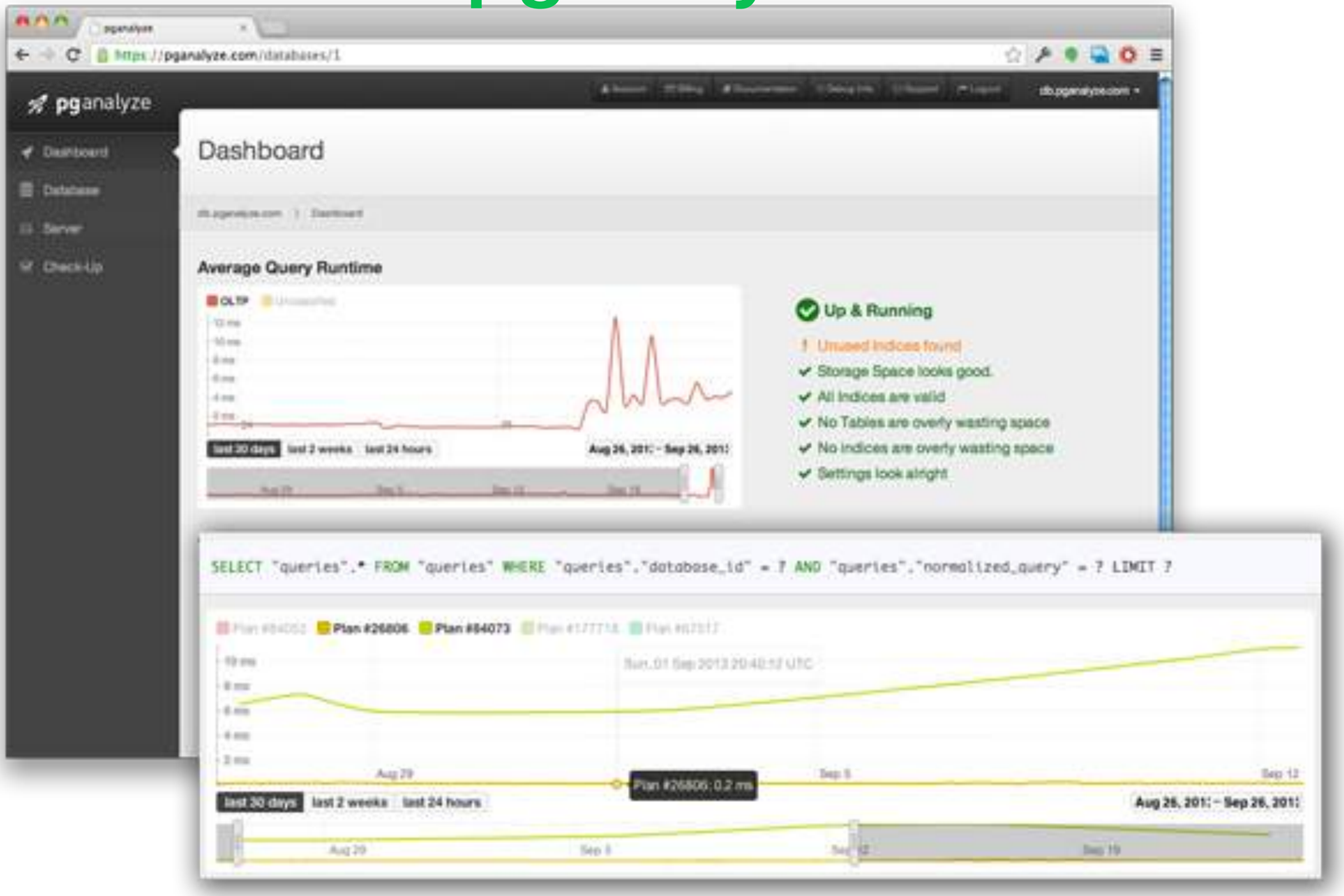
PGObserver



https://github.com/zalando/PGObserver

adyen

pganalyze



https://pganalyze.com

adyen

pgBadger

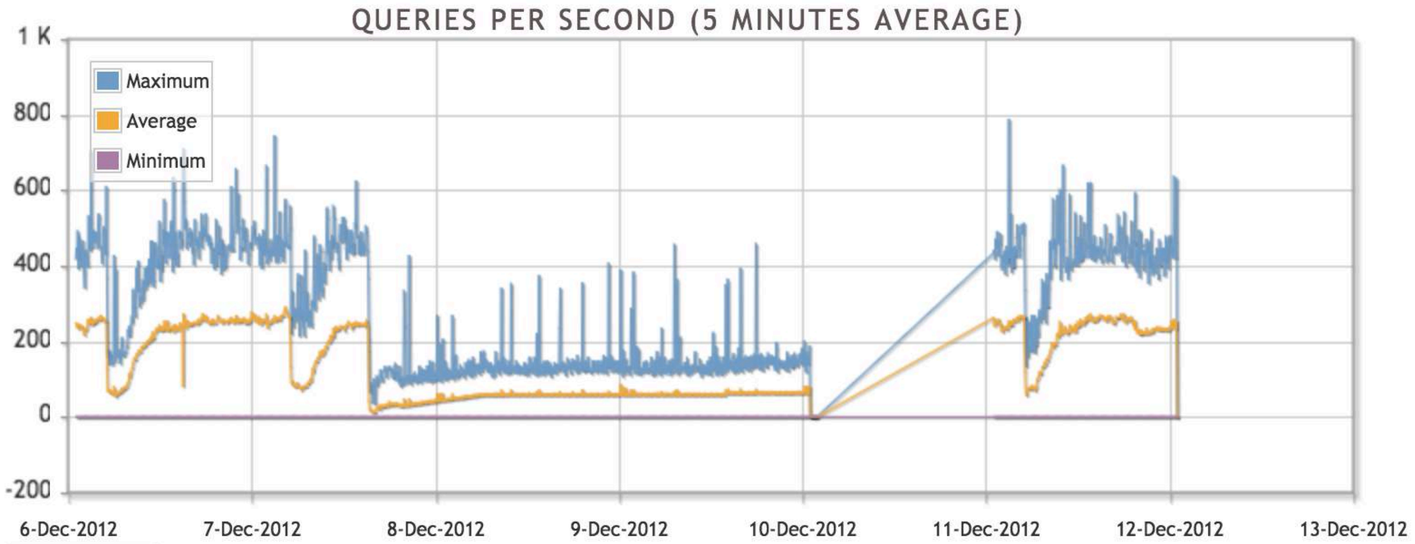
pgBadger Overview Connections Sessions Checkpoints Temp Files Vacuums Locks Queries Top Events PgBouncer

SQL Traffic

KEY VALUES

787 queries/s
Query Peak

2012-12-11 02:02:02
Date



Download

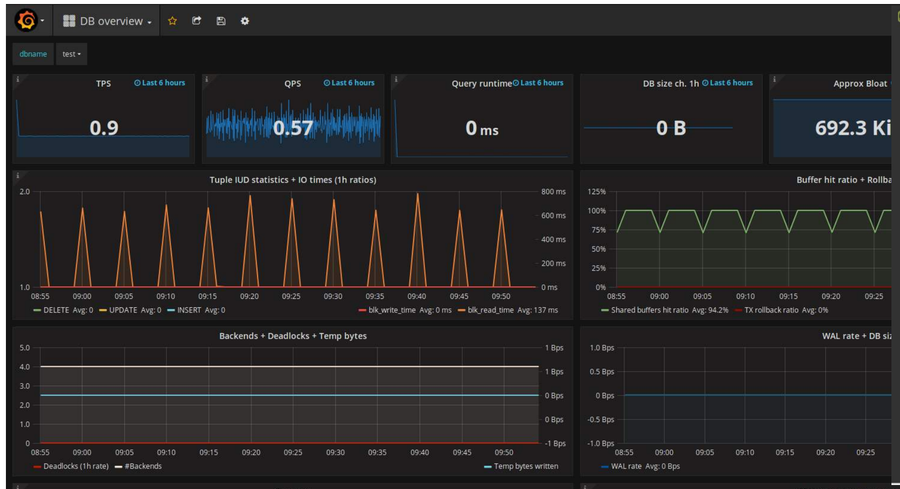
SELECT Traffic

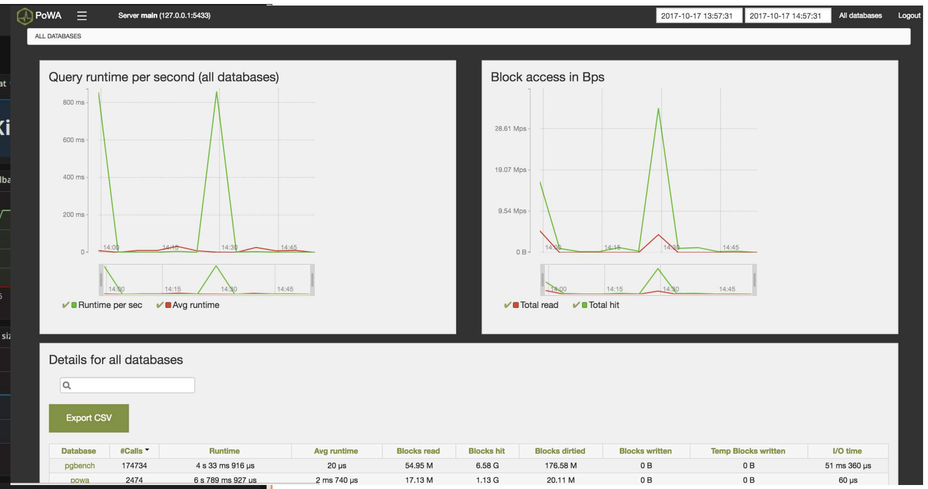
KEY VALUES

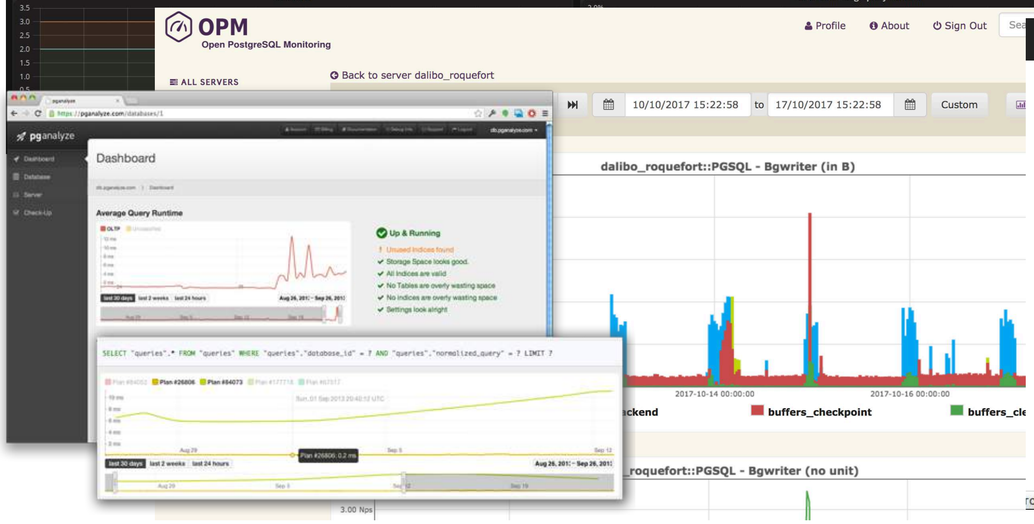
https://github.com/dalibo/pgbadger

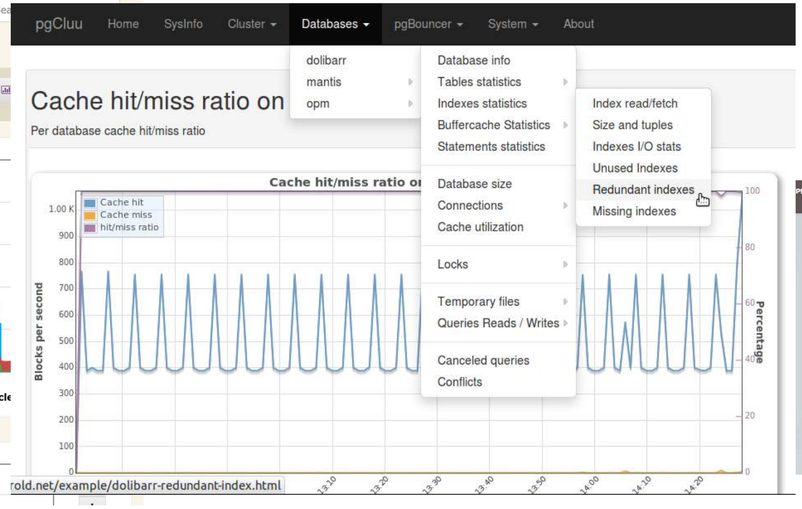
adyen

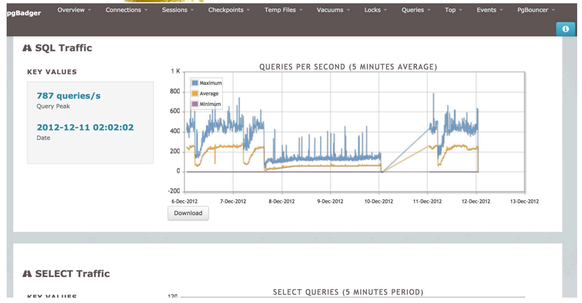
Get a graphing tool



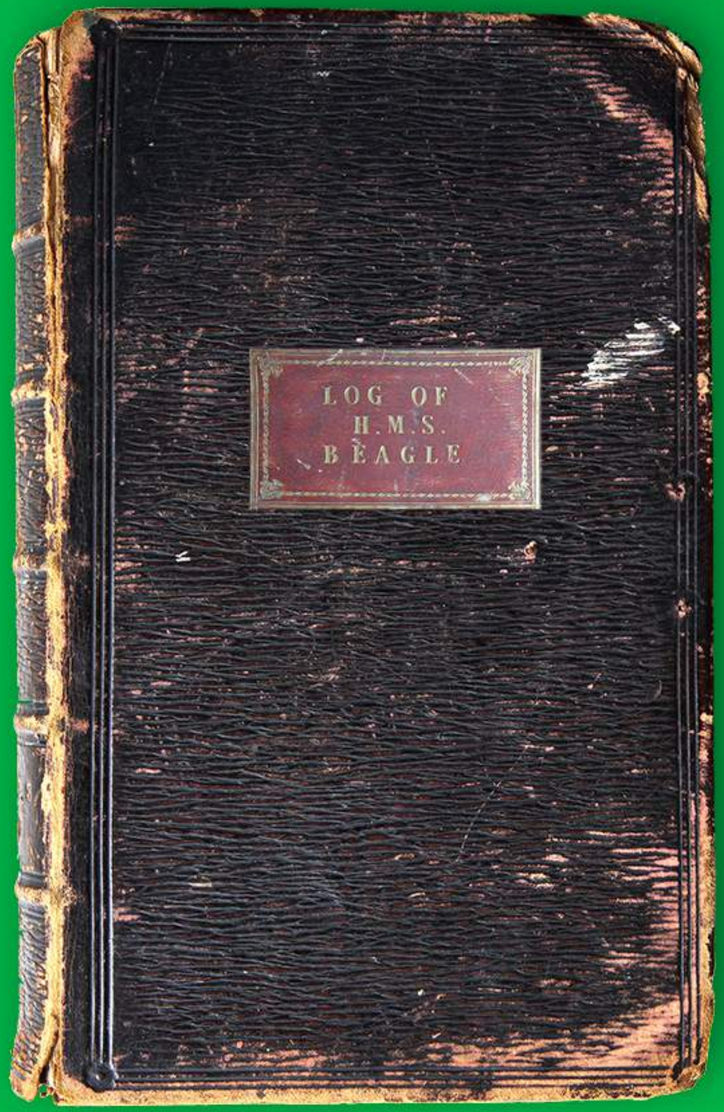




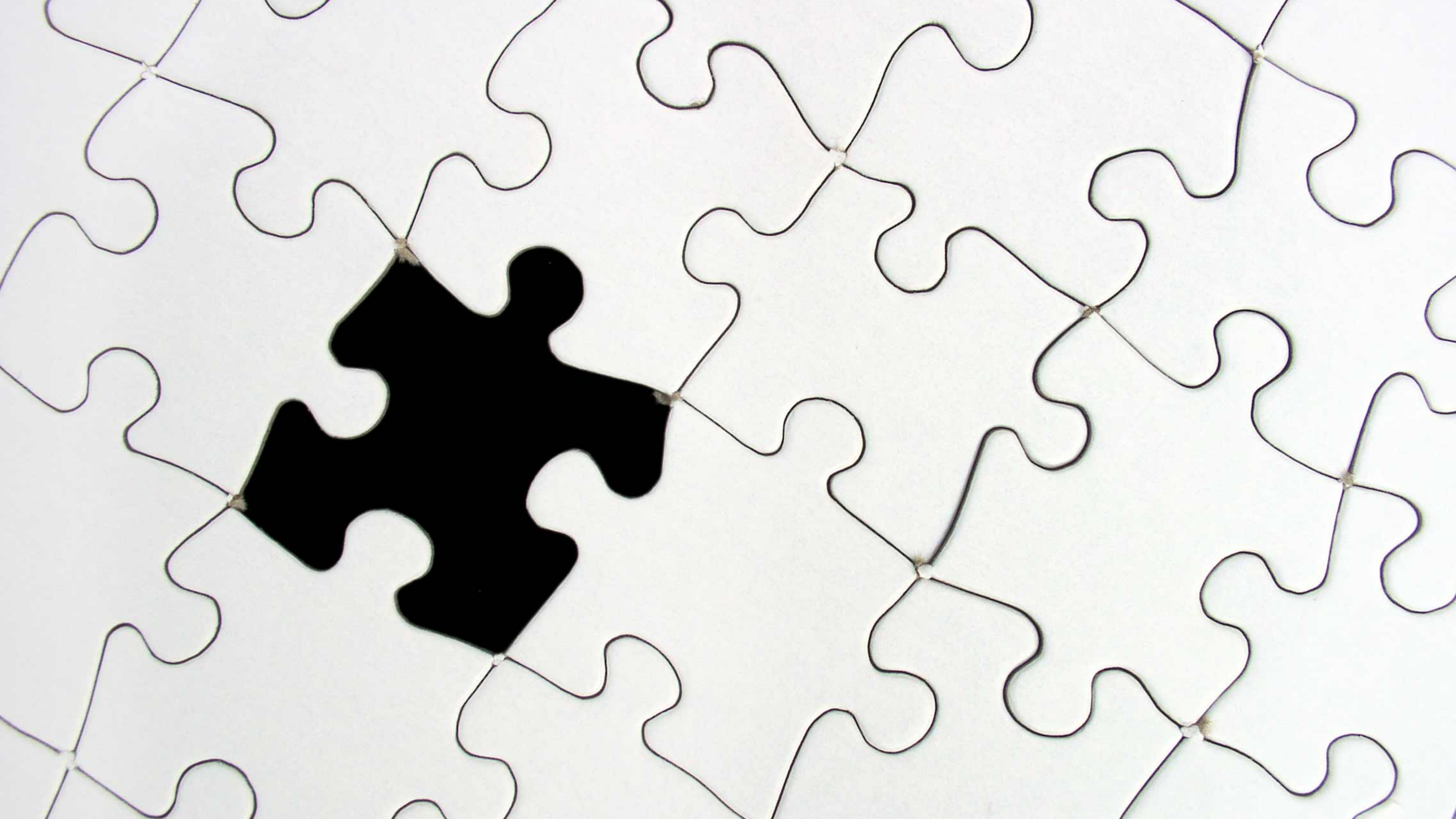




adyen


LOG OF
H.M.S.
BEAGLE


HARDING'S IMPROVED COUNTER
PATENTED.
8326913



The missing piece

60+ TB

adyen

Long running queries

60+ TB

?

?

adyen

> 10,000 tiny queries per second

60+ TB

? ? ? ? ? ? ? ? ? ? ? ?

adyen

pg_stat_statements

```
ALTER SYSTEM SET shared_preload_libraries
    TO ‘pg_stat_statements’;
```

adyen

pg_stat_statements

```
CREATE EXTENSION pg_stat_statements;
```

adyen

Top Query

```
SELECT
    query,
    queryid,
    calls,
    min_time,
    max_time,
    mean_time,
    shared_blks_read,
    shared_blks_hit
FROM
    pg_stat_statements
ORDER BY
    total_time DESC
LIMIT 1;
```

adyen

Query text

```
-[ RECORD 1 ]----+-----------------------------
query            | SELECT *, time::text AS
time_pretty FROM laptimes WHERE $1 = raceid::integer
ORDER BY time ASC LIMIT $2
queryid          | 2617453800
calls            | 155560
min_time         | 29.04867
max_time         | 1073.044577
mean_time        | 81.014002569729 7
shared_blks_read | 1410883
shared_blks_hit  | 417045537
```

How often

```
-[ RECORD 1 ]----+-----------------------------
query            | SELECT *, time::text AS
time_pretty FROM laptimes WHERE $1 = raceid::integer
ORDER BY time ASC LIMIT $2
queryid          | 2617453800
calls            | 155560
min_time         | 29.04867
max_time         | 1073.044577
mean_time        | 81.014002569729 7
shared_blks_read | 1410883
shared_blks_hit  | 417045537
```

adyen

Average time taken

```
-[ RECORD 1 ]----+-----------------------------
query            | SELECT *, time::text AS
time_pretty FROM laptimes WHERE $1 = raceid::integer
ORDER BY time ASC LIMIT $2
queryid          | 2617453800
calls            | 155560
min_time         | 29.04867
max_time         | 1073.044577
mean_time        | 81.0140025697297
shared_blks_read | 1410883
shared_blks_hit  | 417045537
```

adyen

Blocks read from disk

```
-[ RECORD 1 ]----+-----------------------------
query            | SELECT *, time::text AS
time_pretty FROM laptimes WHERE $1 = raceid::integer
ORDER BY time ASC LIMIT $2
queryid          | 2617453800
calls            | 155560
min_time         | 29.04867
max_time         | 1073.044577
mean_time        | 81.014002569729 7
shared_blks_read | 1410883
shared_blks_hit  | 417045537
```

adyen

Blocks hit in shared_buffers

```
-[ RECORD 1 ]----+-----------------------------
query            | SELECT *, time::text AS
time_pretty FROM laptimes WHERE $1 = raceid::integer
ORDER BY time ASC LIMIT $2
queryid          | 2617453800
calls            | 155560
min_time         | 29.04867
max_time         | 1073.044577
mean_time        | 81.014002569729 7
shared_blks_read | 1410883
shared_blks_hit  | 417045537
```

adyen

Query placeholders

```
-[ RECORD 1 ]----+-----------------------------
query            | SELECT *, time::text AS
time_pretty FROM laptimes WHERE $1 = raceid::integer
ORDER BY time ASC LIMIT $2
queryid          | 2617453800
calls            | 155560
min_time         | 29.04867
max_time         | 1073.044577
mean_time        | 81.014002569729 7
shared_blks_read | 1410883
shared_blks_hit  | 417045537
```

adyen

Prepare placeholder query

```
-[ RECORD 1 ]----+-----------------------------
query            | SELECT *, time::text AS
time_pretty FROM laptimes WHERE $1 = raceid::integer
ORDER BY time ASC LIMIT $2
queryid          | 2617453800
calls            | 155560
min_time         | 29.04867
max_time         | 1073.044577
mean_time        | 81.0140025697297
shared_blks_read | 1410883
shared_blks_hit  | 417045537
```

Copy

```
-[ RECORD 1 ]----+-------------------------------
query            | SELECT *, time::text AS
time_pretty FROM laptimes WHERE $1 = raceid::integer
ORDER BY time ASC LIMIT $2
queryid          | 2617453800
calls            | 155560
min_time         | 29.04867
max_time         | 1073.044577
mean_time        | 81.0140025697297
shared_blks_read | 1410883
shared_blks_hit  | 417045537
```

adyen

Paste

```
PREPARE perf1 (integer, integer) AS
                        SELECT *, time::text AS
time_pretty FROM laptimes WHERE $1 = raceid::integer
ORDER BY time ASC LIMIT $2
```

Prepared statement

```
PREPARE perf1 (integer, integer) AS
                        SELECT *, time::text AS
time_pretty FROM laptimes WHERE $1 = raceid::integer
ORDER BY time ASC LIMIT $2
```

Type of placeholders

```
PREPARE perf1 (integer, integer) AS
                        SELECT *, time::text AS
time_pretty FROM laptimes WHERE $1 = raceid::integer
ORDER BY time ASC LIMIT $2
```

Explain prepared query

```
EXPLAIN EXECUTE perf1(1, 1);
```

adyen

Query plan

```
Limit  (cost=9045.26..9045.27 rows=1 width=52)
-> Sort  (cost=9045.26..9050.54 rows=2111 width=52)
     Sort Key: "time"
     -> Seq Scan on laptimes
          (cost=0.00..9034.71 rows=2111 width=52)
          Filter: (1 = (raceid)::integer)
```

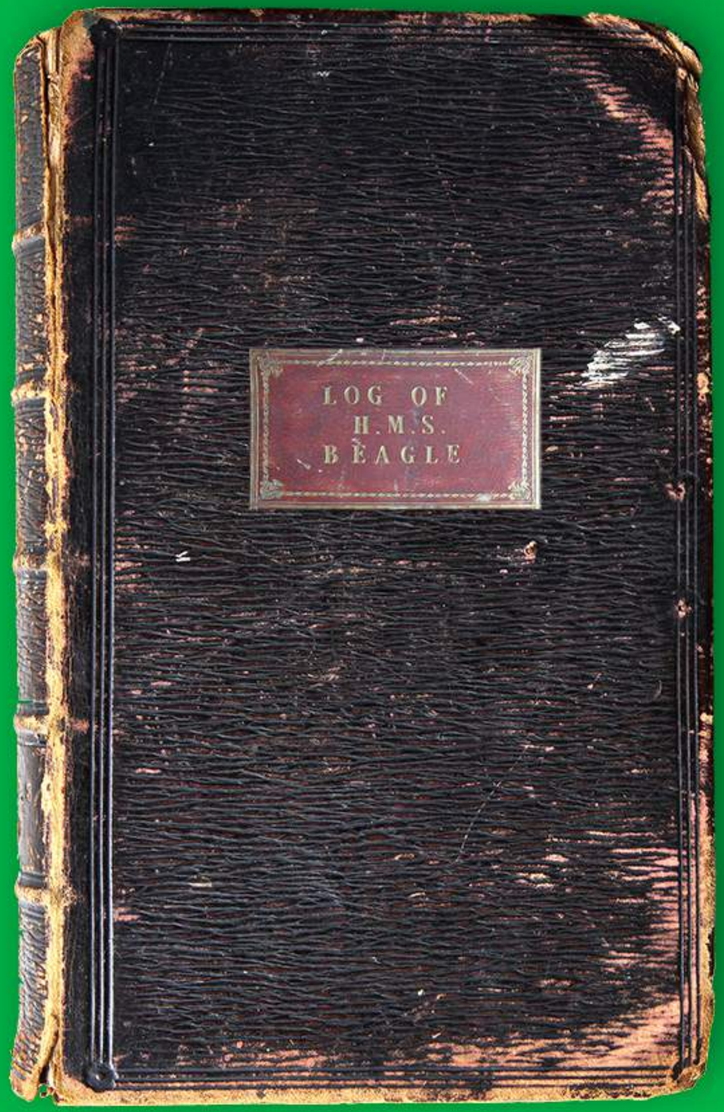
Query plan

```
Limit  (cost=9045.26..9045.27 rows=1 width=52)
-> Sort  (cost=9045.26..9050.54 rows=2111 width=52)
     Sort Key: "time"
     -> Seq Scan on laptimes
          (cost=0.00..9034.71 rows=2111 width=52)
          Filter: (1 = (raceid)::integer)
```

Use

pg_stat_statements

adyen


LOG OF
H.M.S.
BEAGLE


HARDING'S IMPROVED COUNTER
PATENTED.
8326913

OPM
Open PostgreSQL Monitoring

Profile | About | Sign Out | Search amongs servers

ALL SERVERS

DEMO

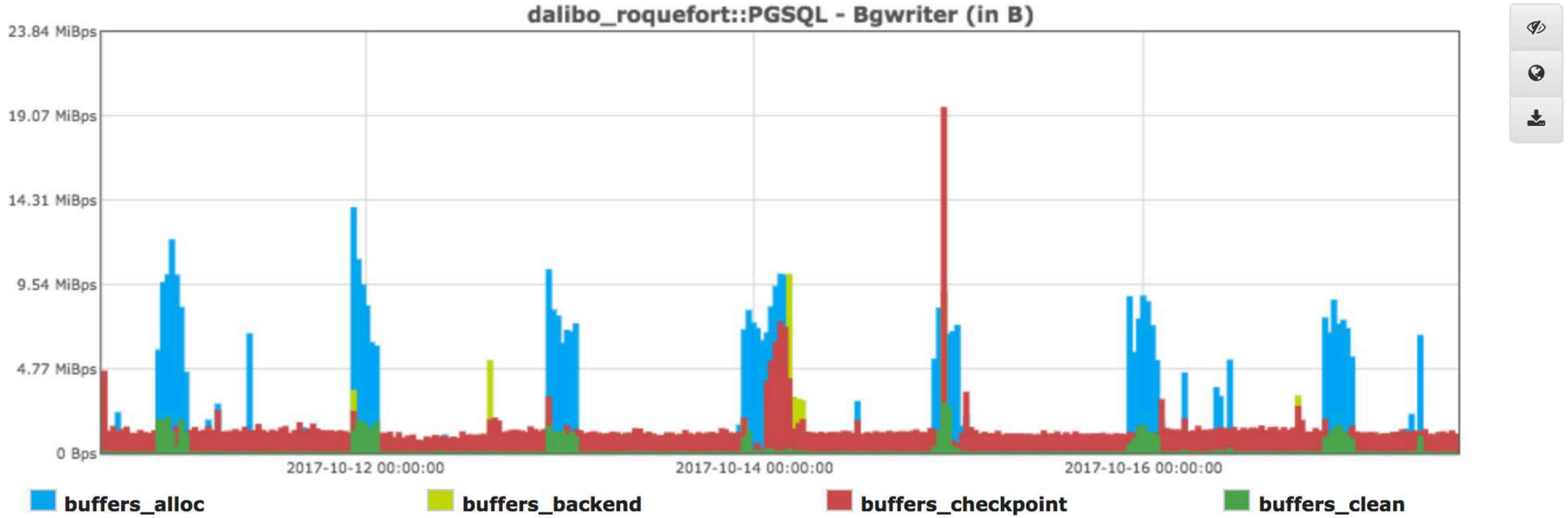
DALIBO_EPOISSES

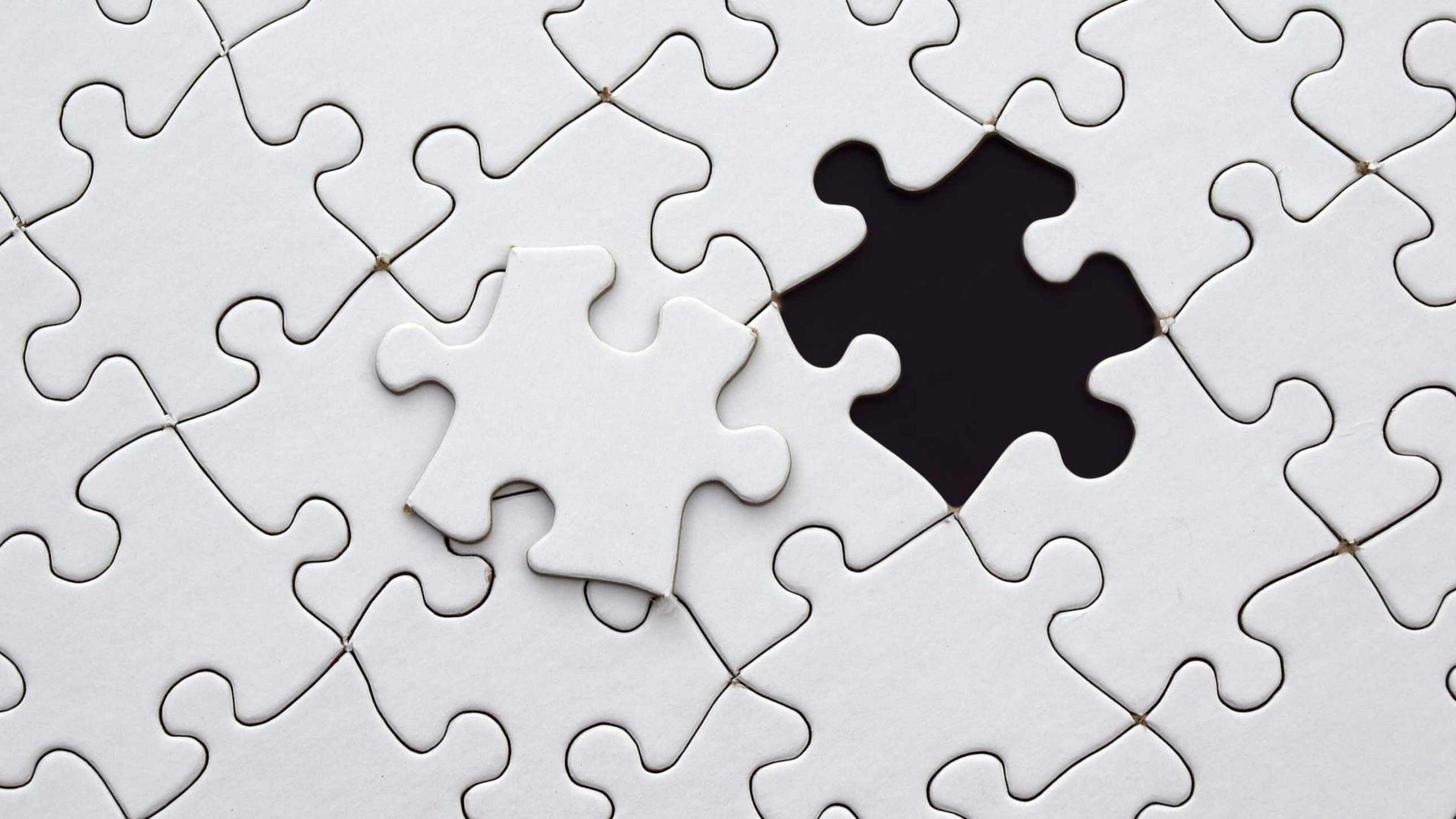
DALIBO_ROQUEFORT

LOCALHOST

Back to server dalibo_roquefort

Year | Month | Week | Day | 10/10/2017 15:22:58 to 17/10/2017 15:22:58 | Custom | Change service...






POSTGRESOL
MAERSK
ESTELLE MÆRSK

THE
LUMBERMAN'S
IMBER MARK GUIDE

Analyzing Database Performance

Feike Steenbergen, PGConf.EU, Warsaw, 2017-10-25