FROM Expensive Queries
TO Smart Caching
A DBA's Guide

by Javier Zon

DISCLOSURE

Who's this guy?

- SysAdmin and MySQL DBA
- Founder of ScaleDB
- @jtomaszon
- Taekwondo Black Belt

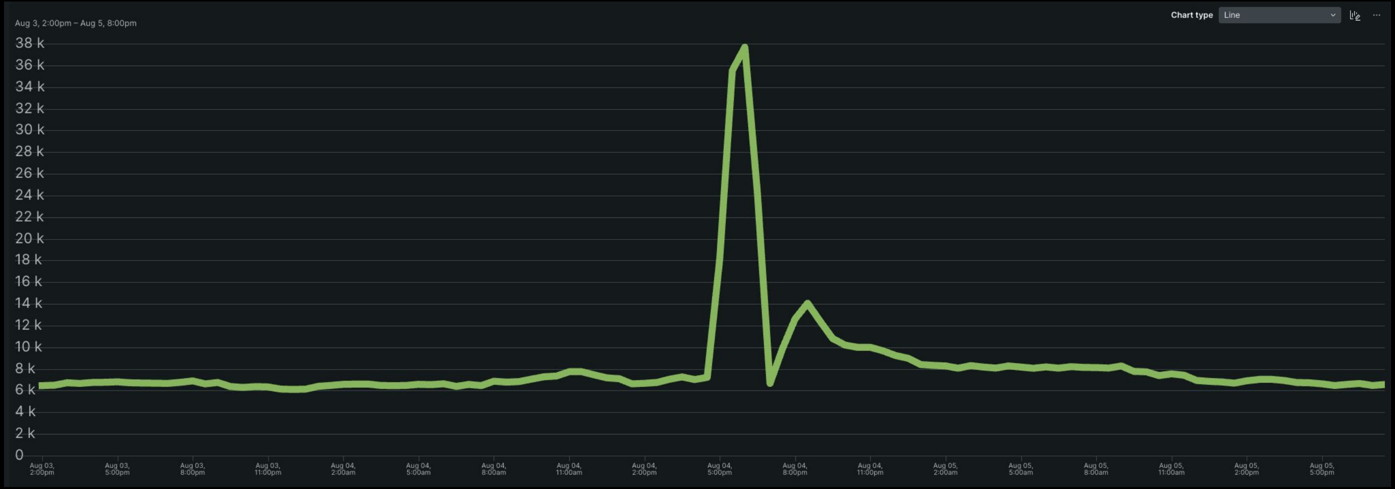
PROBLEM

Application

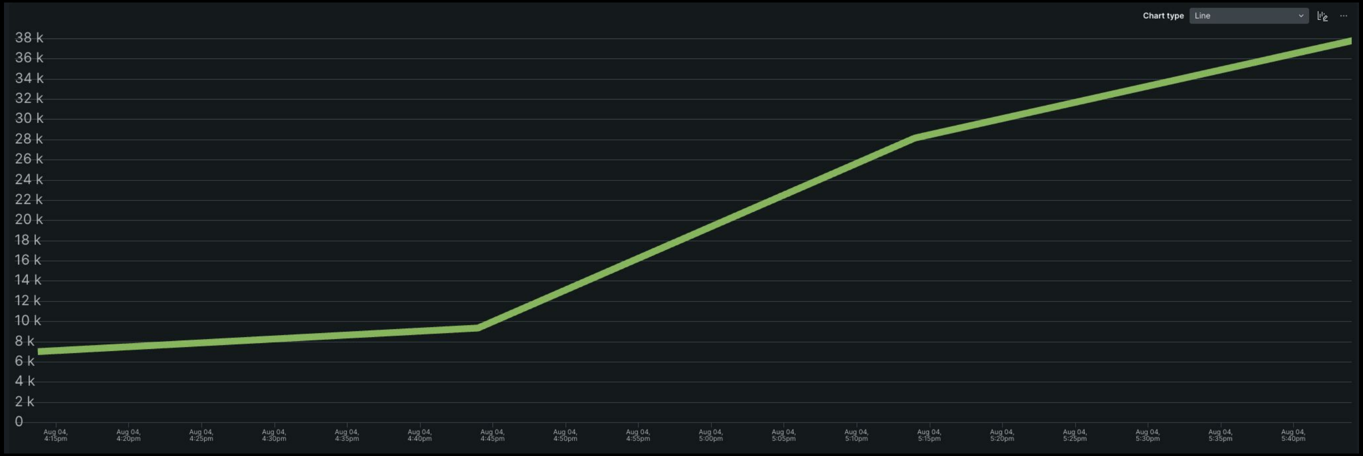
Connection

Pooling

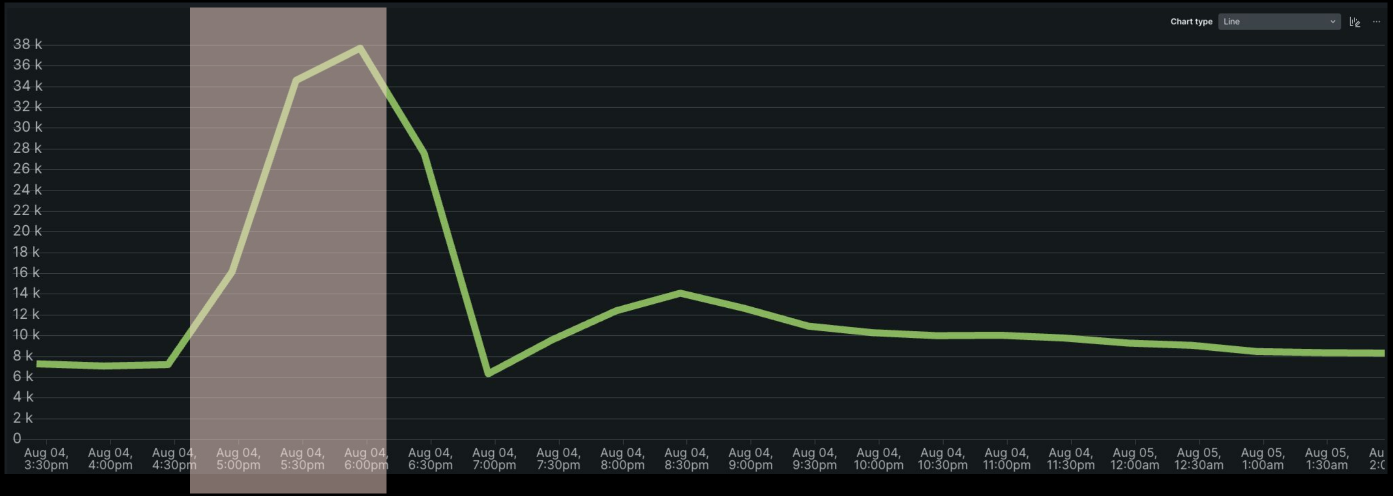
Application Connection Pooling



Application Connection Pooling



Application Connection Pooling



PROBLEM

Object

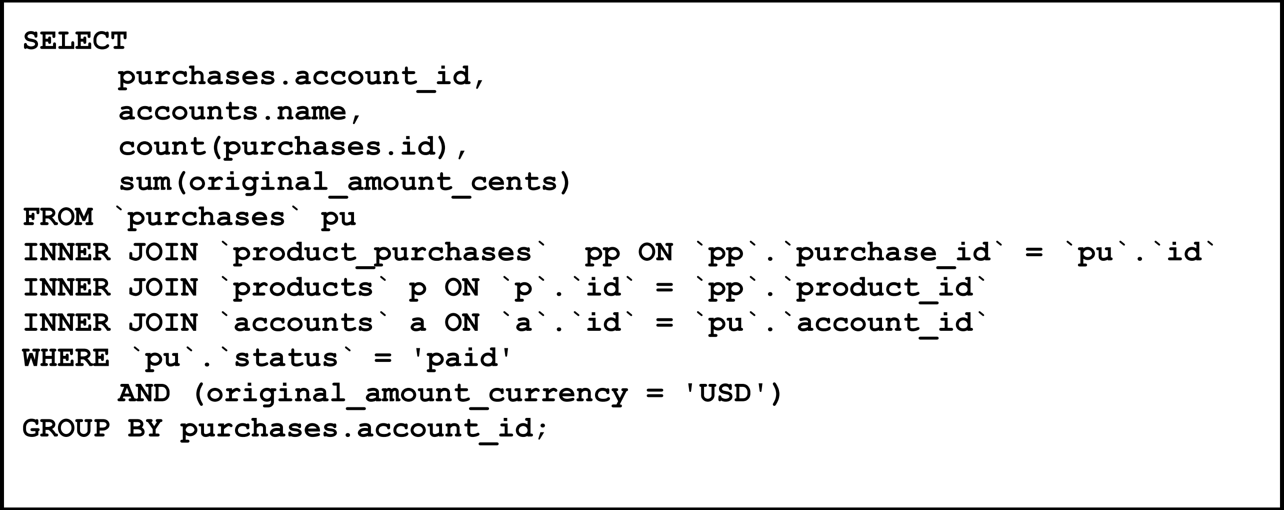
Relational

Mapping "Magic"

Object Relational Mapping "Magic"

```
purchases = Purchase.unscoped.where(status:"paid").
  where("original_amount_currency = ?", "USD").
  joins(:products, :account).
  group("purchases.account_id").
pluck("purchases.account_id",
    "accounts.name",
    "count(purchases.id)",
    "sum(original_amount_cents)"
);
```

Object Relational Mapping "Magic"



```
SELECT
     purchases.account_id,
     accounts.name,
     count(purchases.id),
     sum(original_amount_cents)
FROM `purchases` pu
INNER JOIN `product_purchases`  pp ON `pp`.`purchase_id` = `pu`.`id`
INNER JOIN `products` p ON `p`.`id` = `pp`.`product_id`
INNER JOIN `accounts` a ON `a`.`id` = `pu`.`account_id`
WHERE `pu`.`status` = 'paid'
     AND (original_amount_currency = 'USD')
GROUP BY purchases.account_id;
```

PROBLEM

Manual Code for Cache Invalidation

*There are only two hard things in Computer Science: naming things and **cache invalidation.***

- *Phil Karlton*

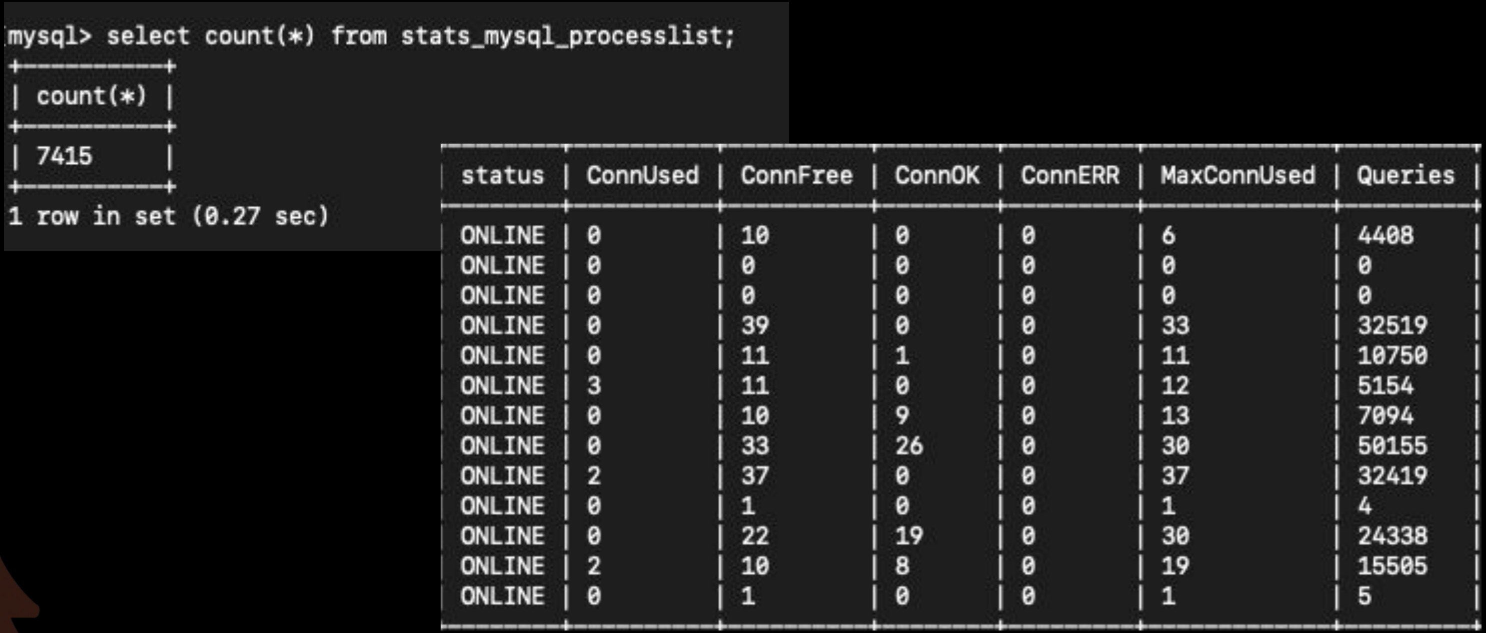
SOLUTION

- ~~Application Connection Pooling~~ > Smarter Connection Pooling
- ~~Object Relational "Magic"~~ > Visibility into Queries
- ~~Cache invalidation~~ > Cache that stays Valid automatically

SOLUTION

- ProxySQL for Connection Pooling
  - Multiplex DB Connections
  - Routing to different Backends

SOLUTION

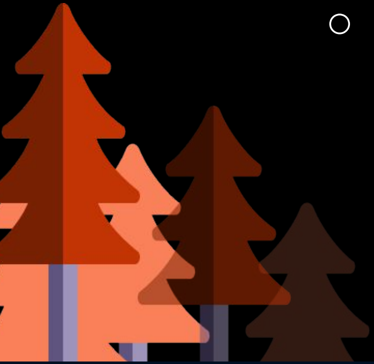


```
mysql> select count(*) from stats_mysql_processlist;
+----------+
| count(*) |
+----------+
| 7415     |
+----------+
1 row in set (0.27 sec)
```

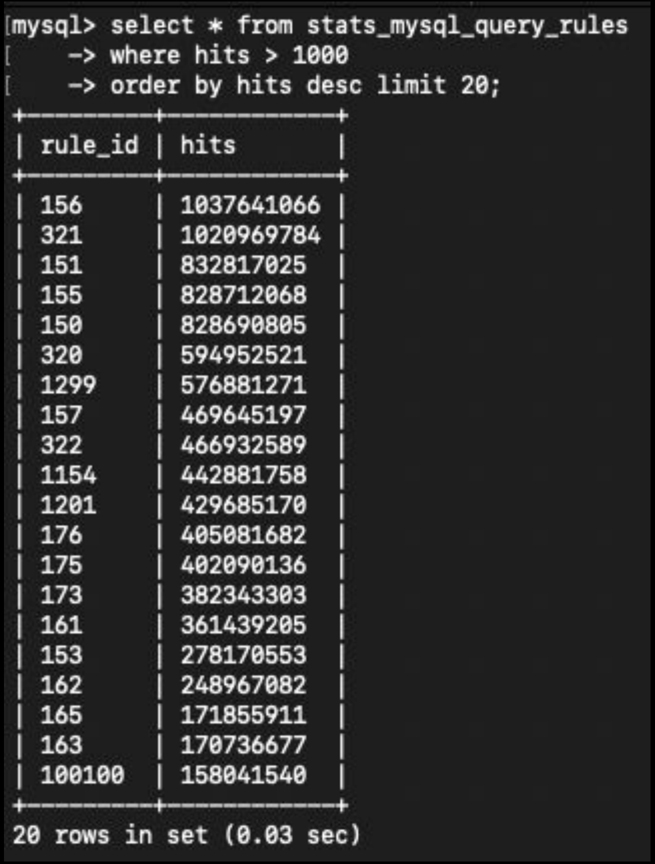
```
+--------+----------+----------+--------+---------+------------+---------+
| status | ConnUsed | ConnFree | ConnOK | ConnERR | MaxConnUsed | Queries |
+--------+----------+----------+--------+---------+------------+---------+
| ONLINE | 0        | 10       | 0      | 0       | 6          | 4408    |
| ONLINE | 0        | 0        | 0      | 0       | 0          | 0       |
| ONLINE | 0        | 0        | 0      | 0       | 0          | 0       |
| ONLINE | 0        | 39       | 0      | 0       | 33         | 32519   |
| ONLINE | 0        | 11       | 1      | 0       | 11         | 10750   |
| ONLINE | 3        | 11       | 0      | 0       | 12         | 5154    |
| ONLINE | 0        | 10       | 9      | 0       | 13         | 7094    |
| ONLINE | 0        | 33       | 26     | 0       | 30         | 50155   |
| ONLINE | 2        | 37       | 0      | 0       | 37         | 32419   |
| ONLINE | 0        | 1        | 0      | 0       | 1          | 4       |
| ONLINE | 0        | 22       | 19     | 0       | 30         | 24338   |
| ONLINE | 2        | 10       | 8      | 0       | 19         | 15505   |
| ONLINE | 0        | 1        | 0      | 0       | 1          | 5       |
+--------+----------+----------+--------+---------+------------+---------+
```

SOLUTION

- ProxySQL "against" ORMMagical
  - "Man-in-the-Middle" Approach
  - Stats on (old/new) Queries
  - The Power of Query Rewrite



SOLUTION



```
[mysql> select * from stats_mysql_query_rules
[    -> where hits > 1000
[    -> order by hits desc limit 20;
+----------+------------+
| rule_id | hits       |
+----------+------------+
| 156     | 1037641066 |
| 321     | 1020969784 |
| 151     | 832817025  |
| 155     | 828712068  |
| 150     | 828690805  |
| 320     | 594952521  |
| 1299    | 576881271  |
| 157     | 469645197  |
| 322     | 466932589  |
| 1154    | 442881758  |
| 1201    | 429685170  |
| 176     | 405081682  |
| 175     | 402090136  |
| 173     | 382343303  |
| 161     | 361439205  |
| 153     | 278170553  |
| 162     | 248967082  |
| 165     | 171855911  |
| 163     | 170736677  |
| 100100  | 158041540  |
+----------+------------+
20 rows in set (0.03 sec)
```

SOLUTION

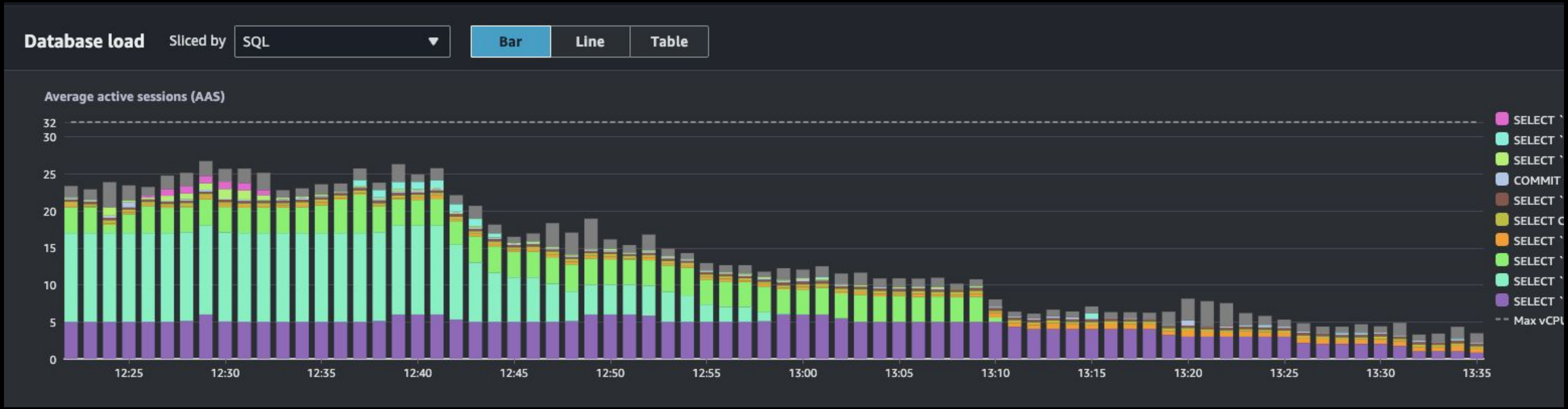
```
SELECT COUNT(*) FROM `users` WHERE `users`.`tenant_id` = ?
AND `users`.`status` = 'active' AND (`users`.`id` IN (
SELECT `users_applied_tags`.`user_id` FROM
`users_applied_tags` WHERE `users_applied_tags`.`tag_id` =
?))
```

```
digest: 0x285F07C6756FC149

match_pattern: ^SELECT.*tenant_id` = (.*)
AND `users`.*tag_id` = (.*)$

replace_pattern: SELECT COUNT(*) FROM `users` u
JOIN users_applied_tags uat ON uat.contact_id = u.id
WHERE u.`tenant_id` = \1 AND u.`status` ='active' AND
uat.`tag_id` = \2
```

SOLUTION

- ReadySet for Smart Caching
  - Precalculated Results
  - NanoSecond response time
  - Automatic cache invalidation via Change Data Capture
  - Failover mechanism

Smart Caching - ReadySet

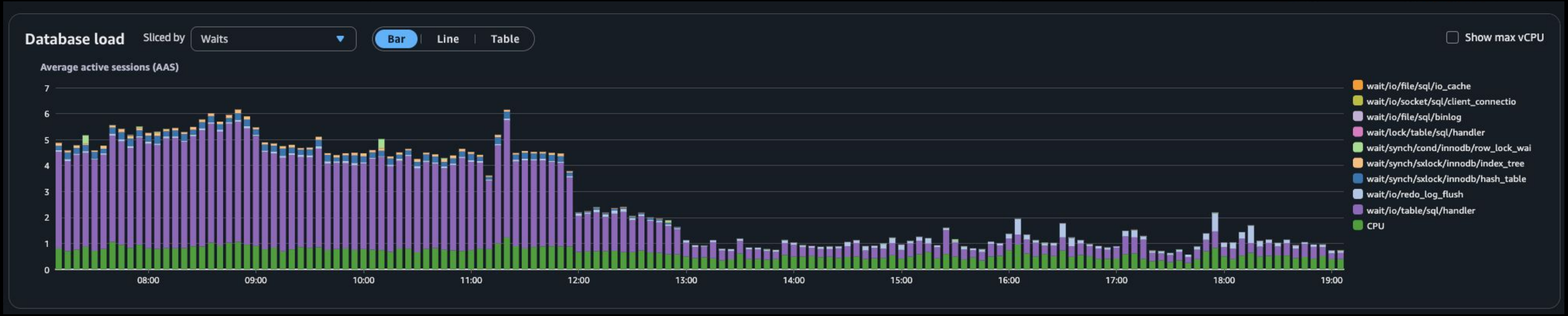


Database load

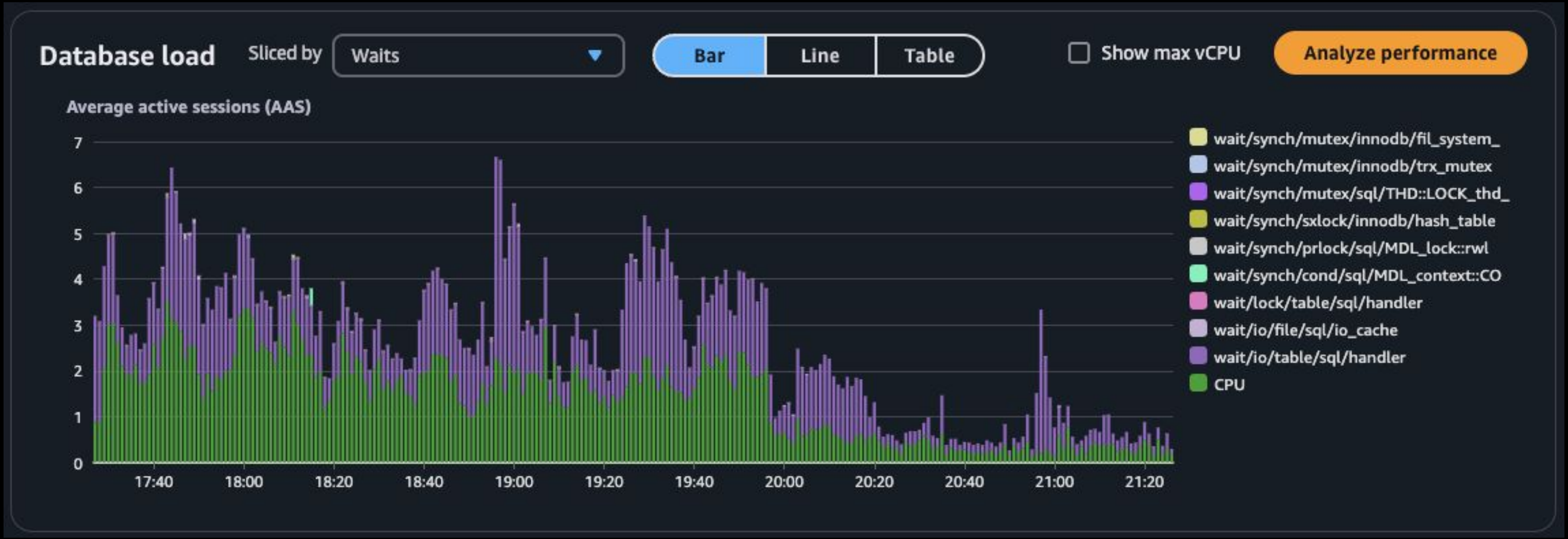
Sliced by: SQL

View: Bar

Smart Caching - ReadySet


Database load
Sliced by
Waits
Bar
Line
Table
□ Show max vCPU
Average active sessions (AAS)
7
6
5
4
3
2
1
0
08:00
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
wait/io/file/sql/io_cache
wait/io/socket/sql/client_connectio
wait/io/file/sql/binlog
wait/lock/table/sql/handler
wait/synch/cond/innodb/row_lock_wai
wait/synch/sxlock/innodb/index_tree
wait/synch/sxlock/innodb/hash_table
wait/io/redo_log_flush
wait/io/table/sql/handler
CPU

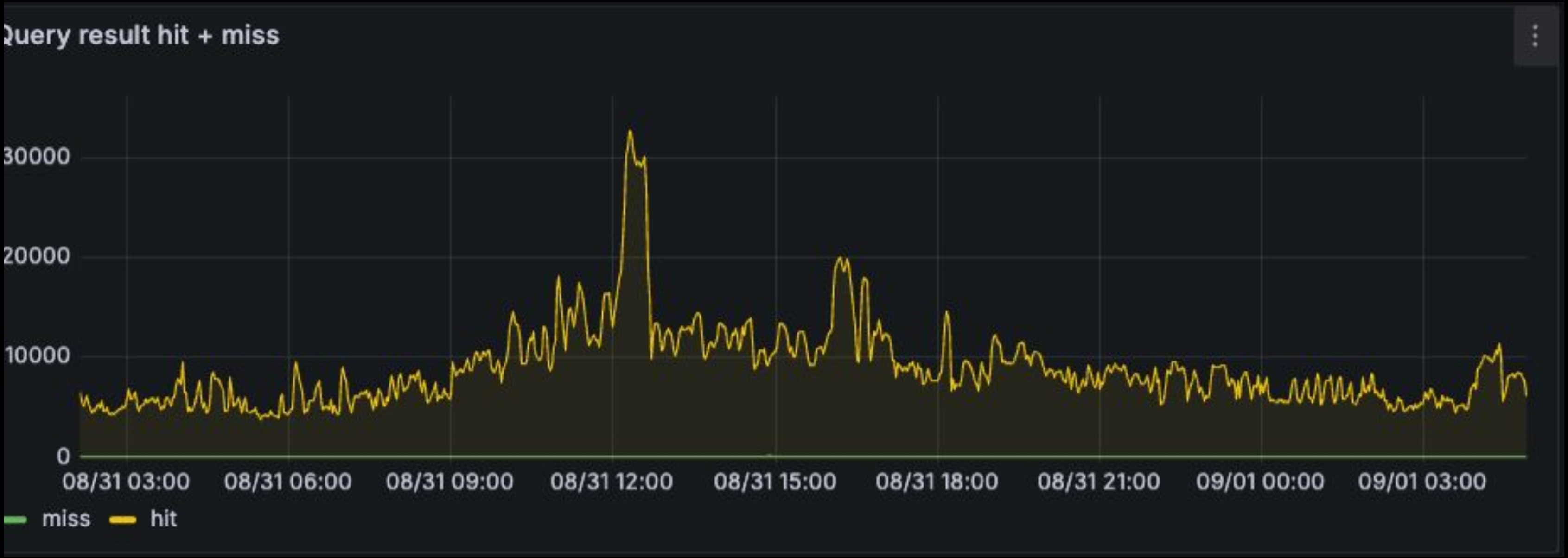
Smart Caching - ReadySet



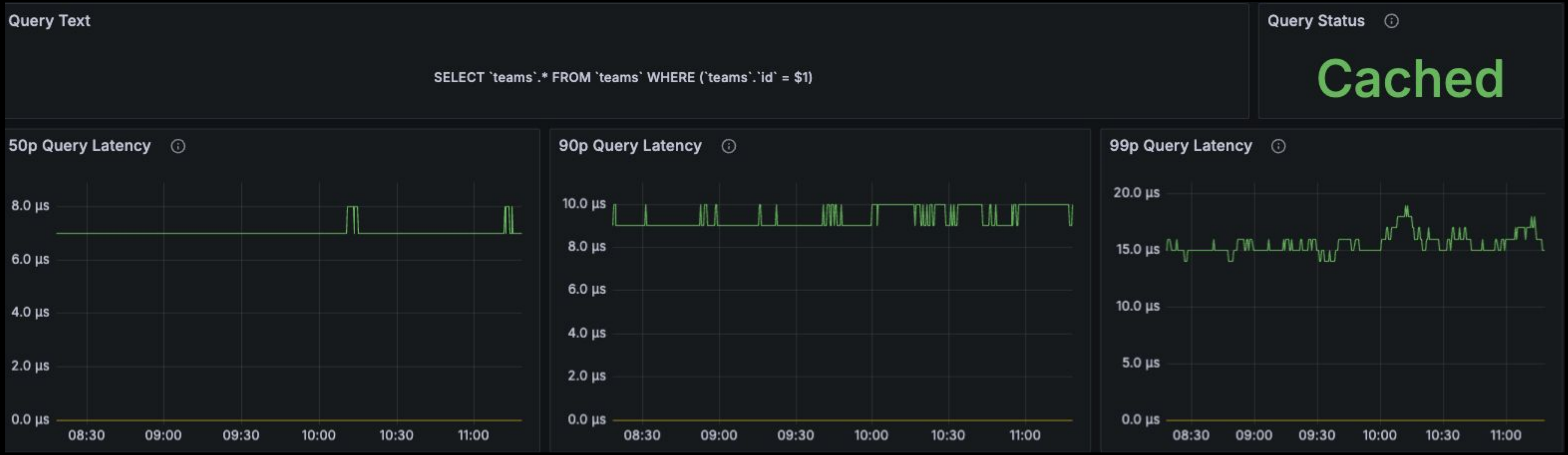
Database load

Sliced by: Waits

Smart Caching - ReadySet



Smart Caching - ReadySet



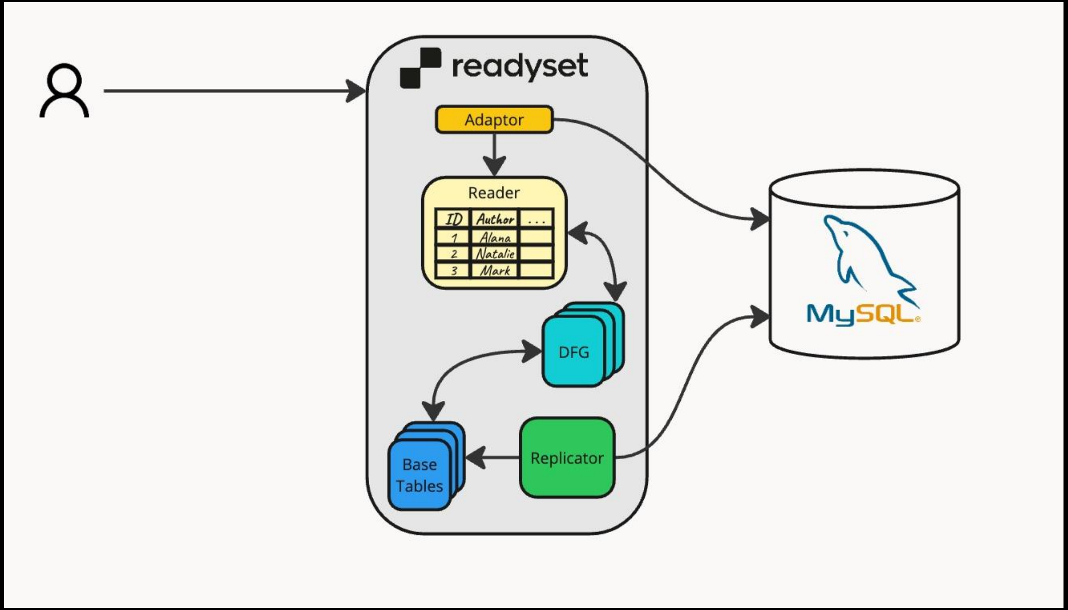
Query Text

```
SELECT `teams`.* FROM `teams` WHERE (`teams`.`id` = $1)
```

Query Status

Cached

Smart Caching - ReadySet



ScaleDB

Questions?

Javier Zon

MySQL Scalability Expert - Taekwondo Instructor