



Slow MySQL database session analysis

6/19/2019

Rolf Martin-Hoster
Staff Engineer



2019

Agenda

- Audience Survey
- Mailchimp
- DBEng at Mailchimp today
- Stone Age
- Bronze Age
- pt-query-digest demo
- Iron Age
- Dark Age
- Modern Age
- Slow query demo
- Computer Age
- Slow session demo
- Wrap up

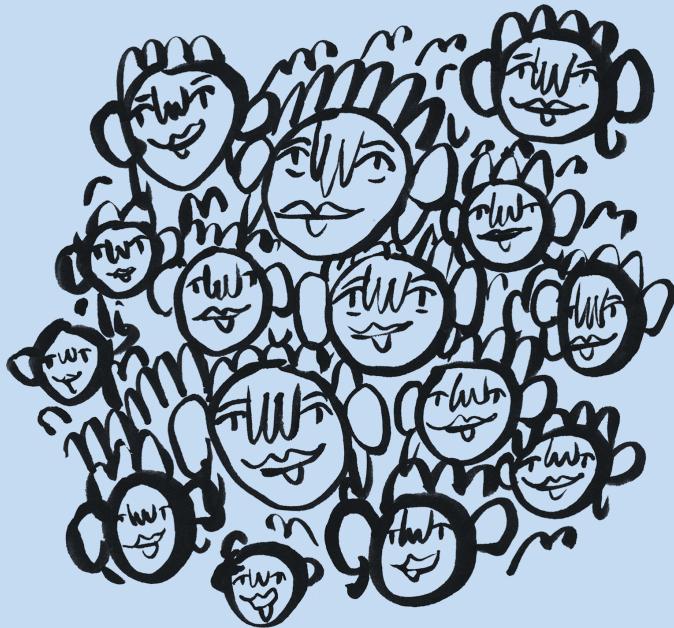


Mailchimp?

“Our all-in-one
Marketing Platform puts
your audience at the
heart of your marketing.”

Freddie
Co-founder, Mailchimp





A story about the evolution of query performance analysis @ MC

265

billion queries per week

247

thousand unique query
fingerprints

365

thousand databases

2200

instances of mysql



The Stone Age



The Stack



The Infrastructure



The Bronze Age



pt-query-digest (aka mk-query-digest)

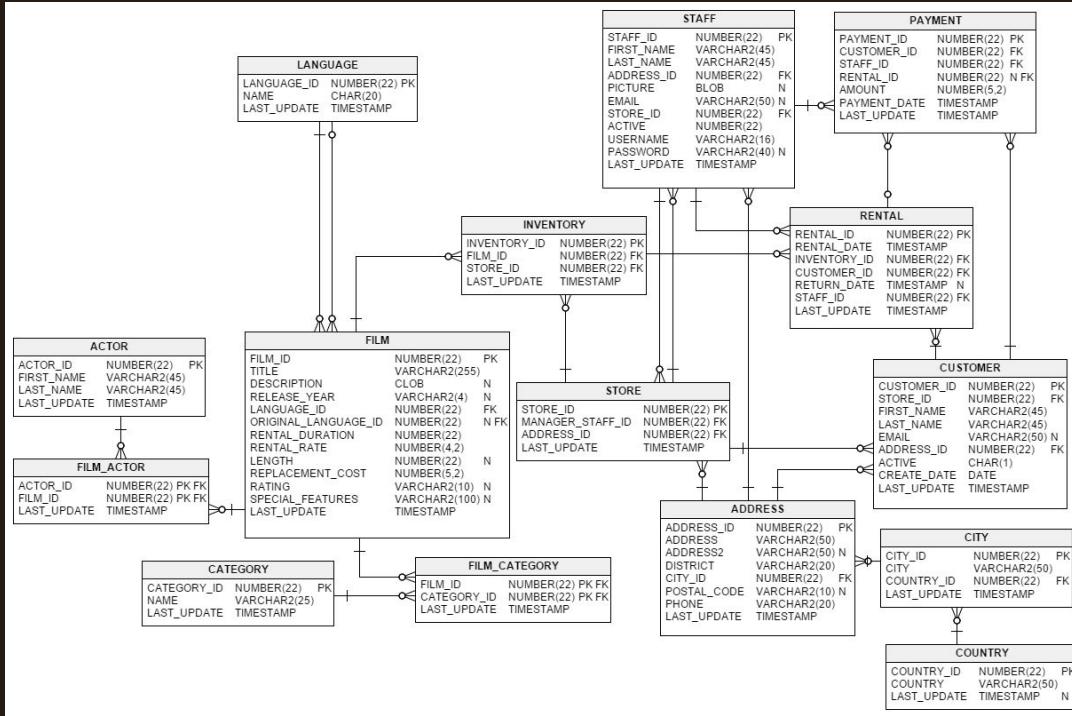
```
# 140ms user time, 10ms system time, 26.18M rss, 220.53M vsz
# Current date: Mon Jun 10 18:49:17 2019
# Hostname: localhost.localdomain
# Files: /var/lib/mysql/localhost-slow.log
# Overall: 55 total, 18 unique, 9.17 QPS, 0.89x concurrency _____
# Time range: 2019-06-10T19:30:21 to 2019-06-10T19:30:27
# Attribute      total     min     max    avg   95% stddev median
# ======      ======  ======  ======  ======  ======  ======  ======
# Exec time        5s    3us    1s   97ms   241ms   277ms   12ms
# Lock time       7ms     0  537us  133us  332us  127us   76us
# Rows sent   399.91k     0  67.19k   7.27k  62.55k  19.95k   0.99
# Rows examine   6.80M     0   1.64M  126.67k 136.54k 367.32k   9.80k
# Rows affecte     3     0     1   0.05     0   0.23     0
# Bytes sent    10.16M     0   2.58M 189.08k 717.31k 591.46k 158.58
# Query size      5.88k    27   378 109.56  246.02   92.54   59.77

# Profile
# Rank Query ID          Response time Calls R/Call V/M   I
# ====== ======          ======  ======  ======  ======  ======  ======
# 1 0xEBD62D27AE01E4002B545F7A875...  3.8154 71.2%    3 1.2718  0.00 SELECT film inventory
# 2 0xA0FF2361631799F4023B57B89F7...  0.6174 11.5%    3 0.2058  0.01 SELECT inventory store film rental
# 3 0x150D10F555208ECCCFE41831920...  0.2927  5.5%    3 0.0976  0.01 SELECT actor film_actor
# 4 0xC5FE76253F27DE3D25C42B40141...  0.1799  3.4%    3 0.0600  0.00 SELECT actor
# 5 0x8F7367E63C50073E7118C6BF199...  0.1474  2.8%    3 0.0491  0.00 SELECT actor
# 6 0x8312E82C71ADB5CDFC0E6554E75...  0.0760  1.4%    3 0.0253  0.00 SELECT film film_category category
film
# MISC 0xMISC           0.2269  4.2%    37 0.0061  0.0 <12 ITEMS>

# Query 1: 0.75 QPS, 0.95x concurrency, ID 0xEBD62D27AE01E4002B545F7A8756BE63 at byte 6509
# Scores: V/M = 0.00
# Time range: 2019-06-10T19:30:23 to 2019-06-10T19:30:27
:|
```



Sakila Sample Database



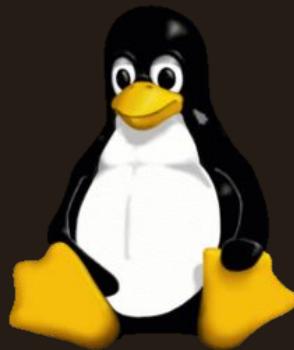
\$> pt-query-digest Demo



The Iron Age



The Stack



The Infrastructure



1.8

billion queries per week

6

thousand unique query
fingerprints

47

thousand databases

200

instances of mysql



The Dark Age



```
$> pt-query-digest --review
```

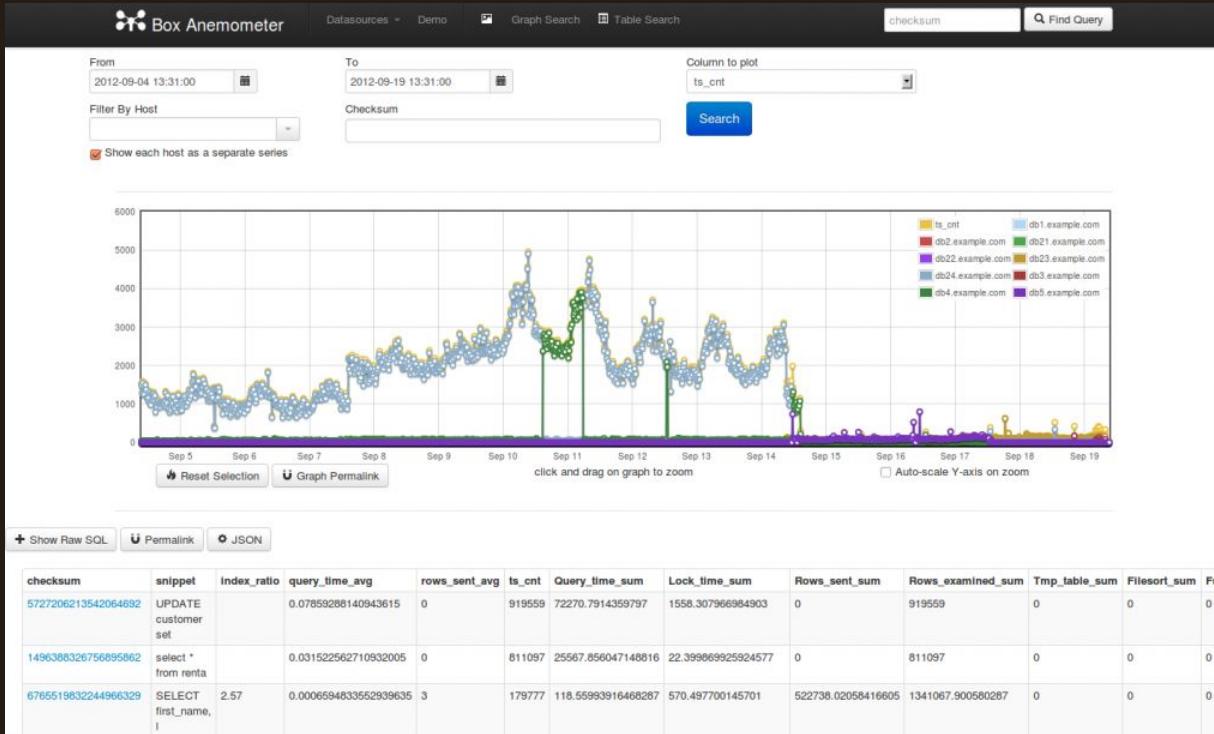
QUERY REVIEW

A query `--review` is the process of storing all the query fingerprints analyzed. This has several benefits:

- You can add metadata to classes of queries, such as marking them for follow-up, adding notes to queries, or marking them with an issue ID for your issue tracking system.
- You can refer to the stored values on subsequent runs so you'll know whether you've seen a query before. This can help you cut down on duplicated work.
- You can store historical data such as the row count, query times, and generally anything you can see in the report.



Box's Anemometer



Box's Anemometer Now Demonetized

box / Anemometer

Watch ▾ 134 Star 1,196 Fork 285

Code Issues 47 Pull requests 6 Projects 0 Security Insights

Box SQL Slow Query Monitor

208 commits 4 branches 0 releases 25 contributors Apache-2.0

Branch: develop ▾ New pull request Create new file Upload files Find File Clone or download ▾

akuzminsky Ported #137 + test

conf Added localhost as seperate datasource file that will be loaded autom... 4 years ago

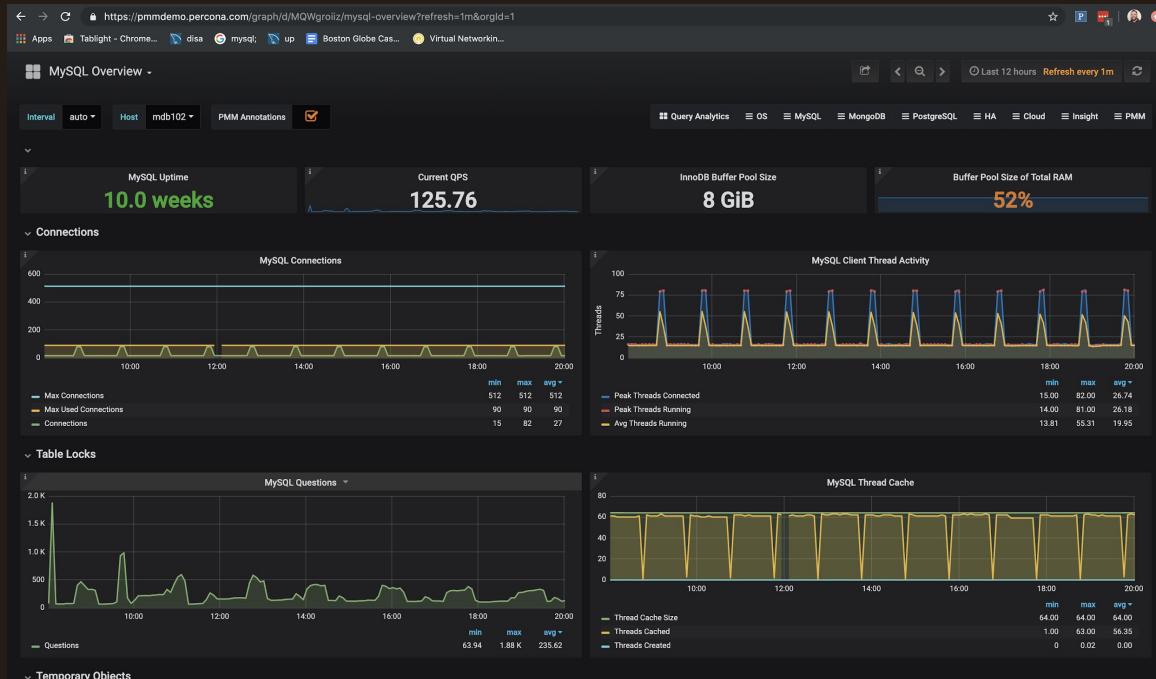
css Made datasource list scrollable (fixes #161) 3 years ago

Latest commit 1a4cca9 on Oct 12, 2016

Author	Commit Message	Date
akuzminsky	Ported #137 + test	Oct 12, 2016
conf	Added localhost as seperate datasource file that will be loaded autom...	4 years ago
css	Made datasource list scrollable (fixes #161)	3 years ago



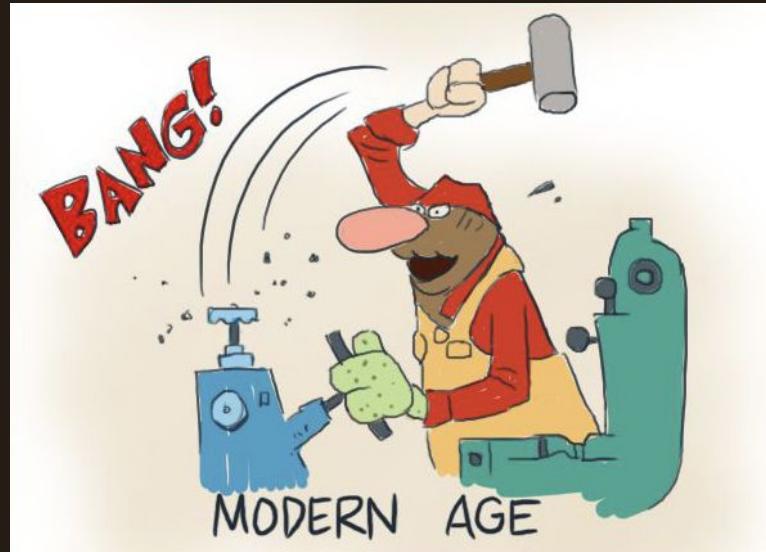
Percona Monitoring and Management



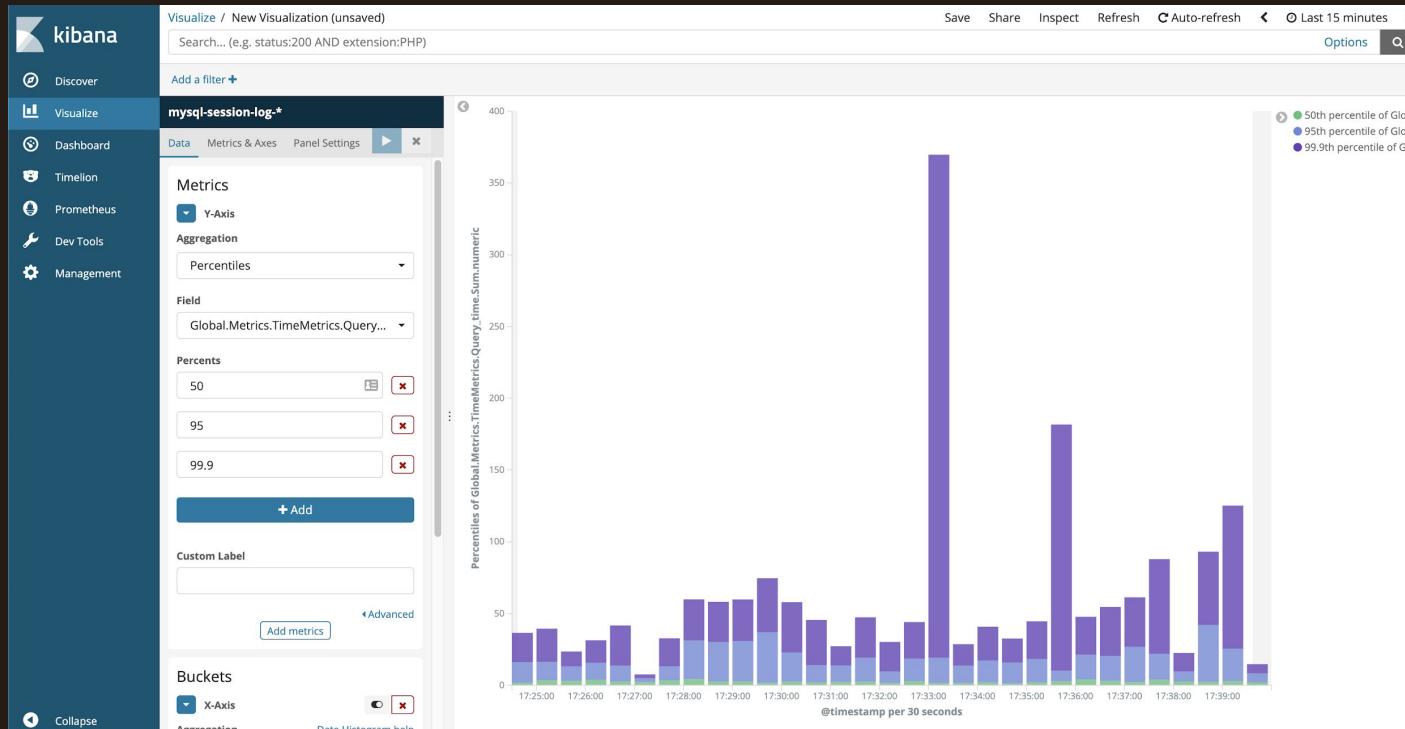
PMM's Limit (back then)



The Modern Age



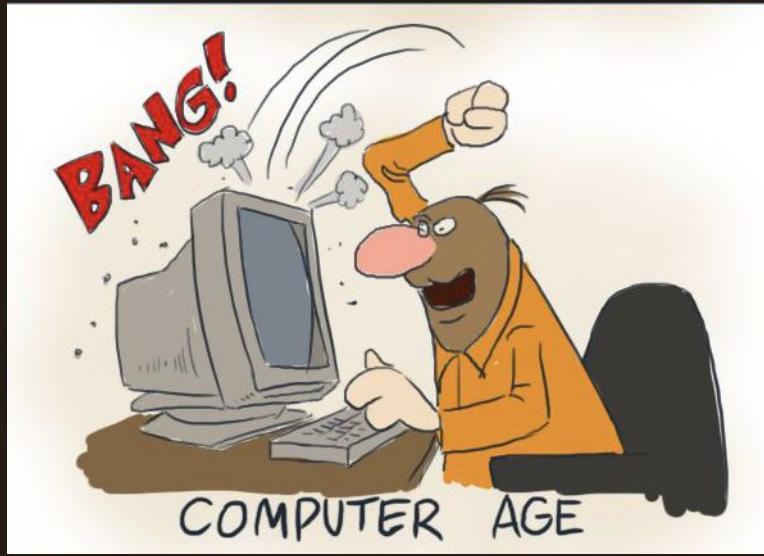
\$> pt-query-digest --output=json > ELK



Slow Query Demo



Computer Age



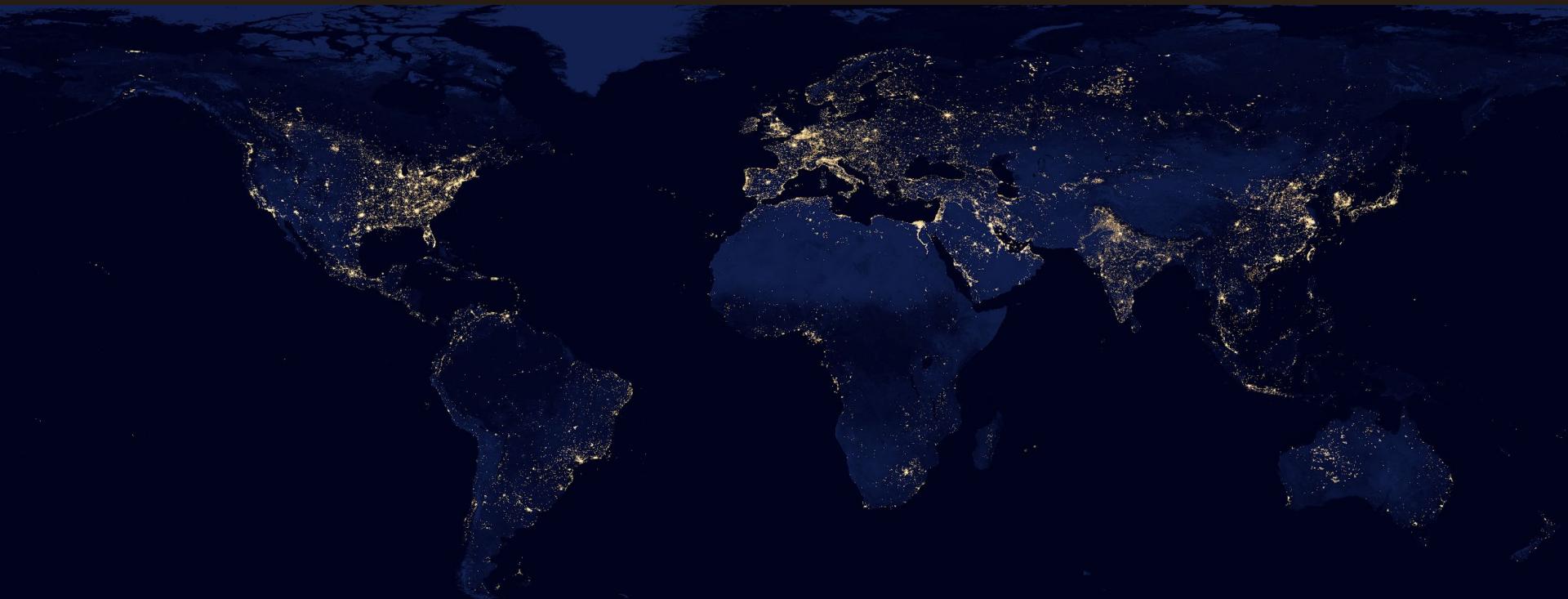
Remember The Infrastructure?



We grew . . . bunches.



More like....



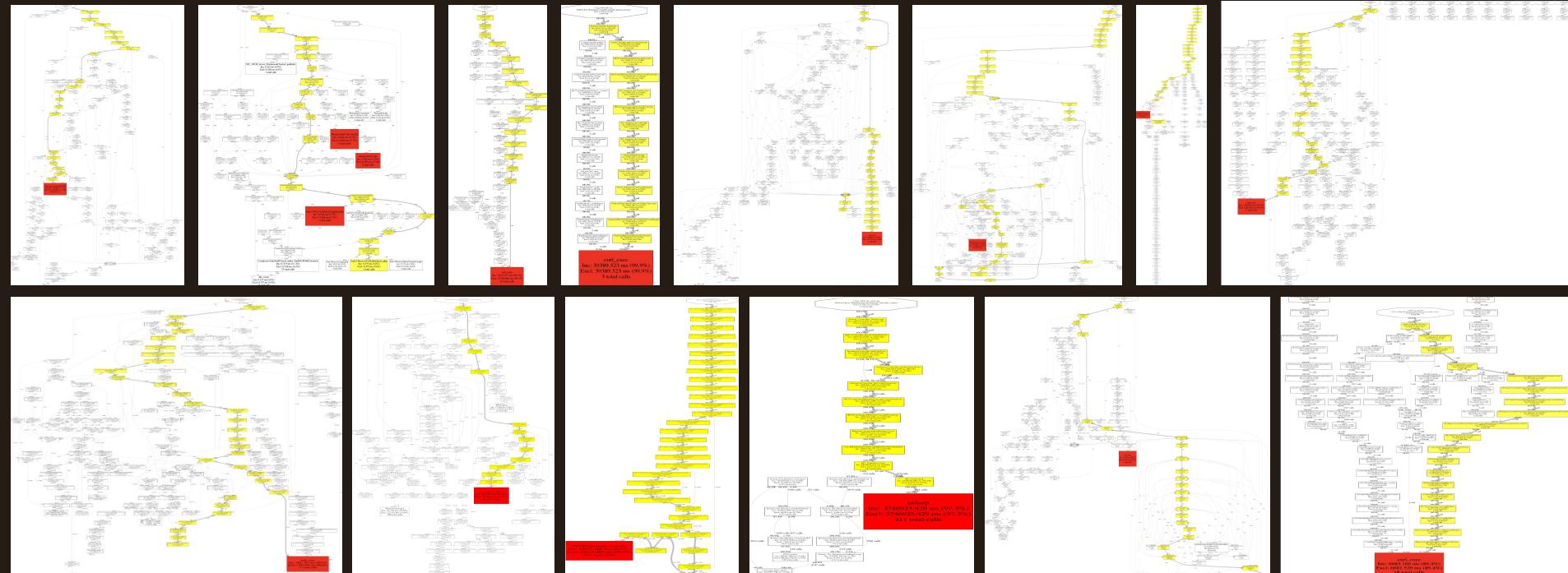
The Code Stack



The Tech Stack



The Call Stacks



265

billion queries per week

247

thousand unique query
fingerprints

365

thousand databases

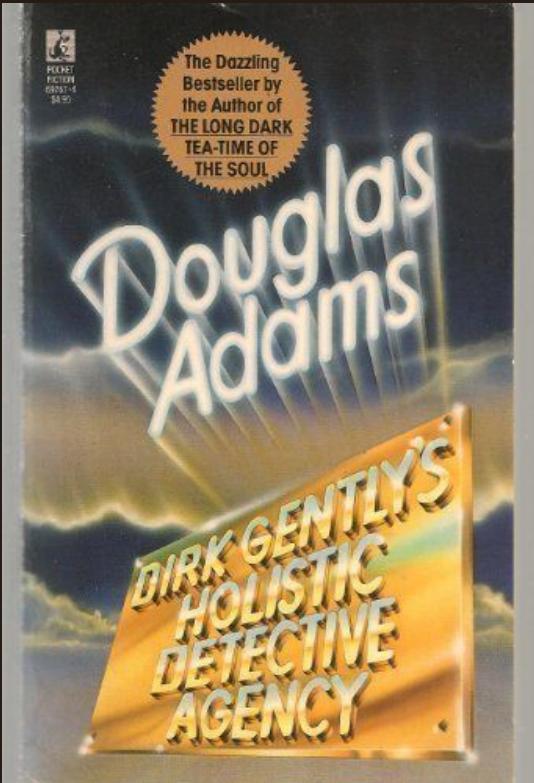
2200

instances of mysql



NOISE
NOISE

A more holistic approach...



We needed to
prioritize database
performance
based on the
customer
experience...



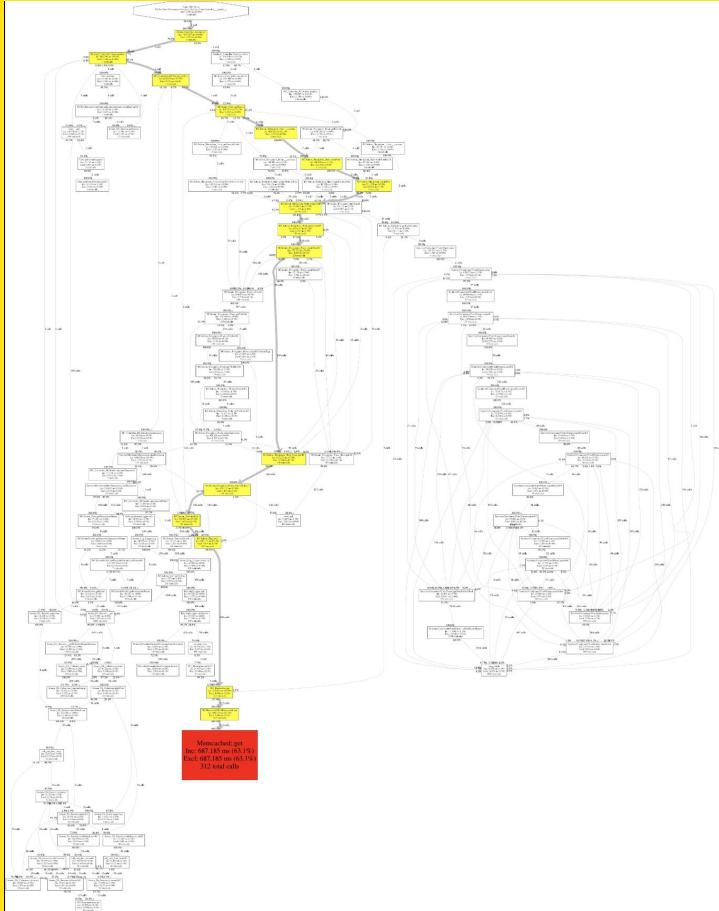
App Side Tracing?



Query Provenance? (query comments)

```
Query
▼ INSERT INTO foo SELECT ... create_more_data.php*/
▼ SELECT * FROM foo ORDER.../*!99999 retrieve.php*/
▼ SELECT * FROM foo ORDER.../*!99999 retrieve.php*/
▼ INSERT INTO foo SELECT ...99999 create_data.php*/
▼ CREATE TABLE foo ( `a` .../*!99999 testcase.php*/
▼ SHOW DATABASES /*!99999 main.php*/
▼ SHOW DATABASES
▼ INSERT INTO foo VALUES ...99999 create_data.php*/
▼ SHOW tables
▼ SELECT @@version_comment LIMIT ?
▼ INSERT INTO foo VALUES ...99999 create_data.php*/
```





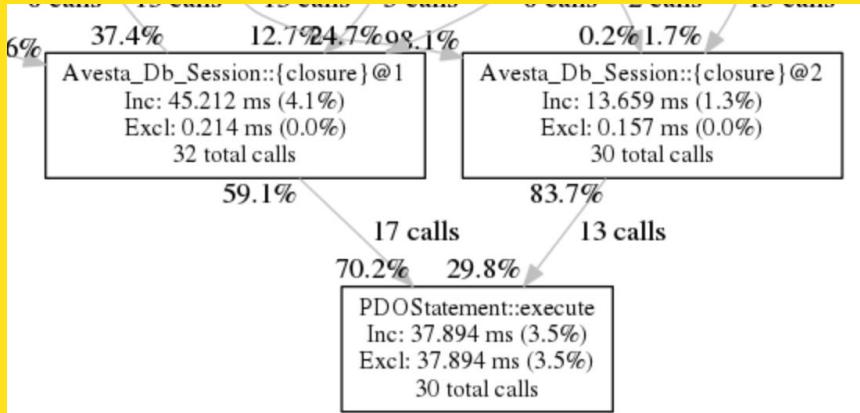
Which file name? Which line?

Which calling parent?

How much of a stack trace?

How would you correlate it?

Have we accomplished anything?



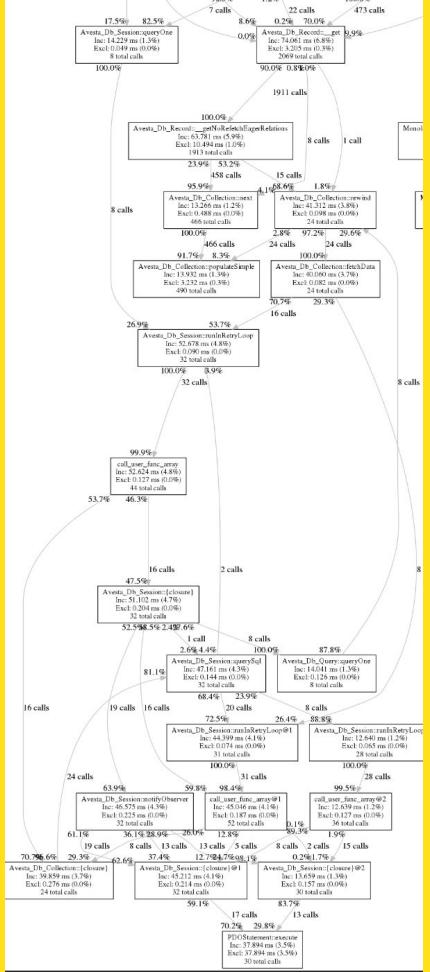
Which file name? Which line?

Which calling parent?

How much of a stack trace?

How would you correlate it?

Have we accomplished anything?



Which file name? Which line?

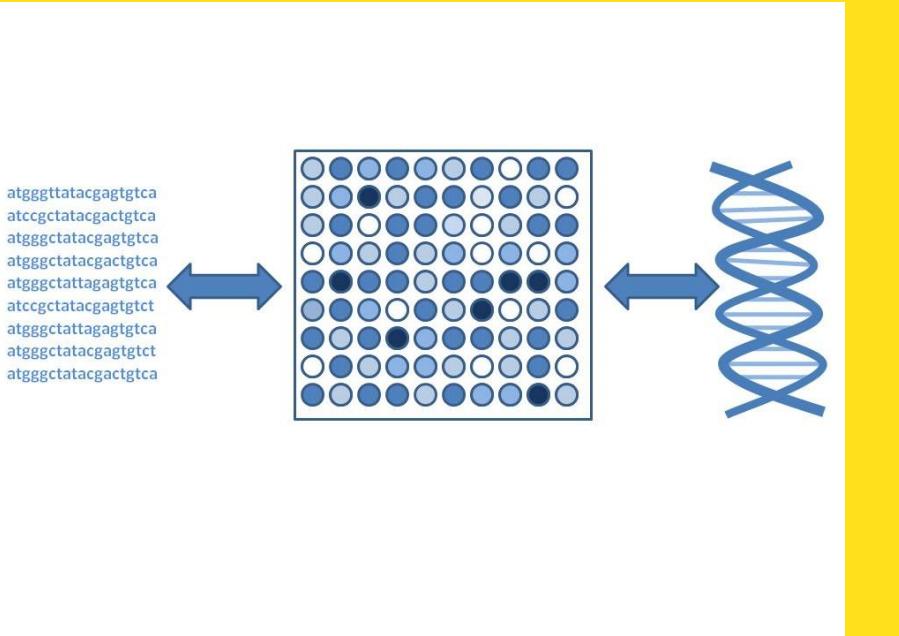
Which calling parent?

How much of a stack trace?

How would you correlate it?

Have we accomplished anything?





Which file name? Which line?

Which calling parent?

How much of a stack trace?

How would you correlate it?

Have we accomplished anything?



Which file name? Which line?

Which calling parent?

How much of a stack trace?

How would you correlate it?

Have we accomplished anything?

Inspiration

The screenshot shows the official website for the Sixteenth Annual Southern California Linux Expo (SCALE 16x). The header features a blue penguin logo and the text "SCALE 16x" followed by "The Sixteenth Annual Southern California Linux Expo". A navigation bar below includes links for Schedule, Speakers, Presentations, Events, Exhibitors, Register, CFP, and Information.

The main content area displays a session titled "Analyzing MySQL Binlogs" by "Franck Levener". Below the title, a brief description of the topic is provided, mentioning MySQL replication lag and transaction analysis. A profile picture of Franck Levener is shown to the right of the text.

Sponsor logos for Diamond, Platinum, and Gold sponsors are displayed on the right side of the page. The Diamond Sponsors include a green Android-like robot icon. The Platinum Sponsors include Microsoft. The Gold Sponsors include mediatemple and MySQL.

Session details listed on the page include:

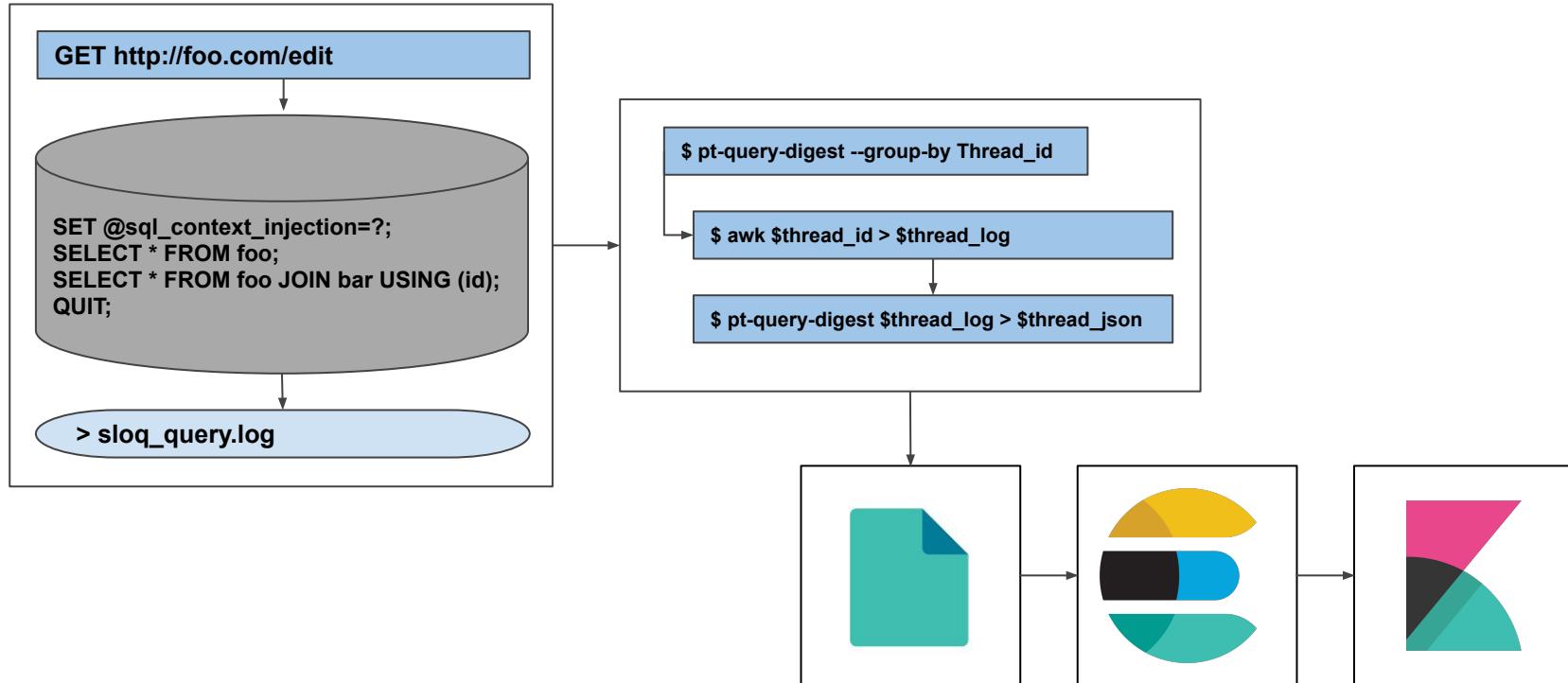
- Topic:** MySQL
- Description:** How many transaction does the database handle per table, per hour? Which table has the most inserts? Does it affect replication lag? Do we delete data from this table? Which hour has the highest peak of updates, and on which tables? By parsing MySQL binlogs , analyzing them, and summarizing them in a table, not only you can answers all these questions but you can also quickly find out potential transactions that can affect Mysql replication lag.
- Room:** Room 101
- Time:** Friday, March 9, 2018 - 15:00 to 16:00



Enlightenment!

The entirety of a user's experience is
that of a single session,
not a single query.

Session Analysis v1



Slow Session Demo





Thank you.



winmutt @ <https://mysqlcommunity.slack.com>



<https://github.com/winmutt>



@atwinmutt



<https://www.linkedin.com/in/rolfm2/>

