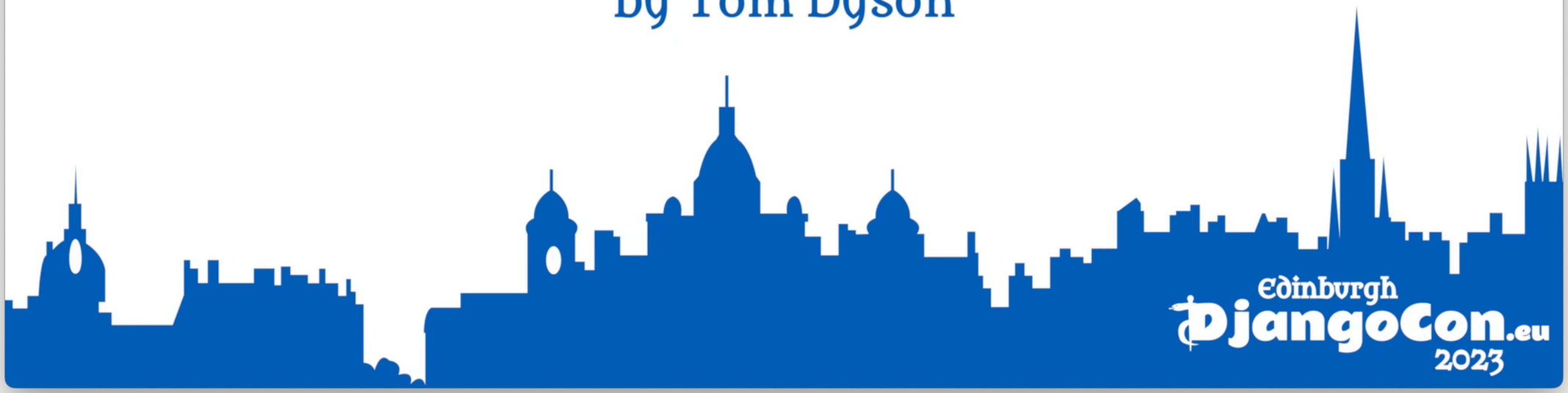
Django, SQLite, and Production

DjangoCon Europe

Anže Pečar, June 7 2024

Use SQLite in production

by Tom Dyson



SQLite in production?

It depends

What is SQLite?

fedidevs.com

FEDIDEVS

Accounts | Conferences

*Discover **10341** superb devs from across the Fediverse!*

Filter By

Filter by name, bio, or instance... Go

Programming Language

Python (1397) Rust (772)

JavaScript (711) Ruby (538)

PHP (519) Java (452) Swift (450)

TypeScript (354) C++ (338)

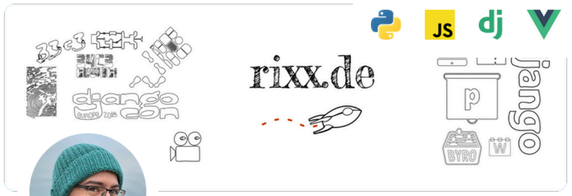
C# (305) CSS (296) R (211)

Go (197) Nix (176) Kotlin (138)

Haskell (98) Scala (60) Julia (45)

F# (40) OCaml (27)

Frameworks, Libraries, and others



Following

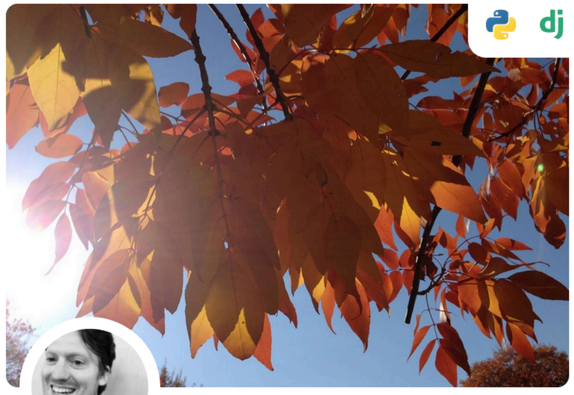
rixx

@rixx

pretalx▪Python/Django/vue.js▪Three Django projects in a trench coat▪he/him

Admin here. Please report instead of DMing.

Last seen	Posts	Followers
5 days, 15 ho...	17,670	5,807



Following

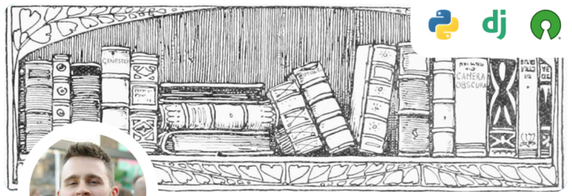
Jeff Triplett

@webology

⛺ @defnado cofounder, 🏝️ @djangocon organizer, 🏎️ @revsys consultant, 🐍 @Thepsf former director, vice-chair, and treasurer, 🦄 @djangoproject member, 🏀, ✨, 💪, ⛹️, 🤖, Oh Mai.

If you are into #django and #python...

Last seen	Posts	Followers
Less than a d...	7,620	1,660



Following

Adam Johnson

@adamchainz

#Django technical board member

🍕 London Django Meetup Organizer with @cgl

✍️ Author of three books on Django and Git

🇬🇧 London / 🇵🇹 Lisbon

Last seen	Posts	Followers
1 day, 15 hour...	2,376	1,336

Following

Following

fedidevs.com

FEDIDEVS

Accounts | Conferences

Programming Language

Python (12) Ruby (6) Swift (3)

JavaScript (2) Rust (2) Golang (1)

CSS (1) PHP (1)

Frameworks, Libraries, and others

Open Source (6) Postgres (4)

Django (2) React (1) Linux (1)

🔴 Live Conferences

Happening right now! You can join the discussion even if you are not attending!

CONFERENCE	LOCATION	POSTS
DjangoCon EU Jun 5 - 9, 2024	Vigo, Spain	127

Upcoming Conferences

Join the discussion and get hyped for the upcoming conferences.

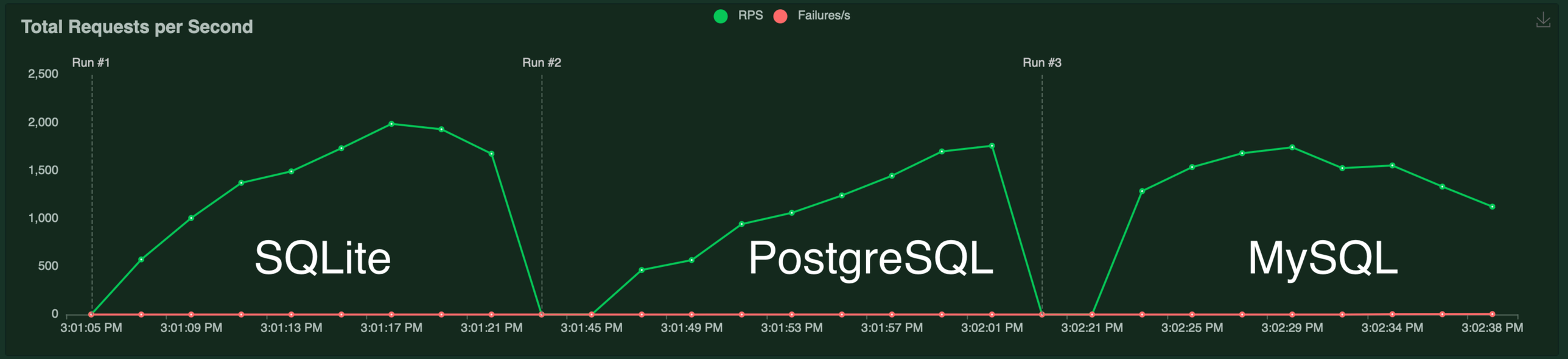
CONFERENCE	LOCATION	POSTS

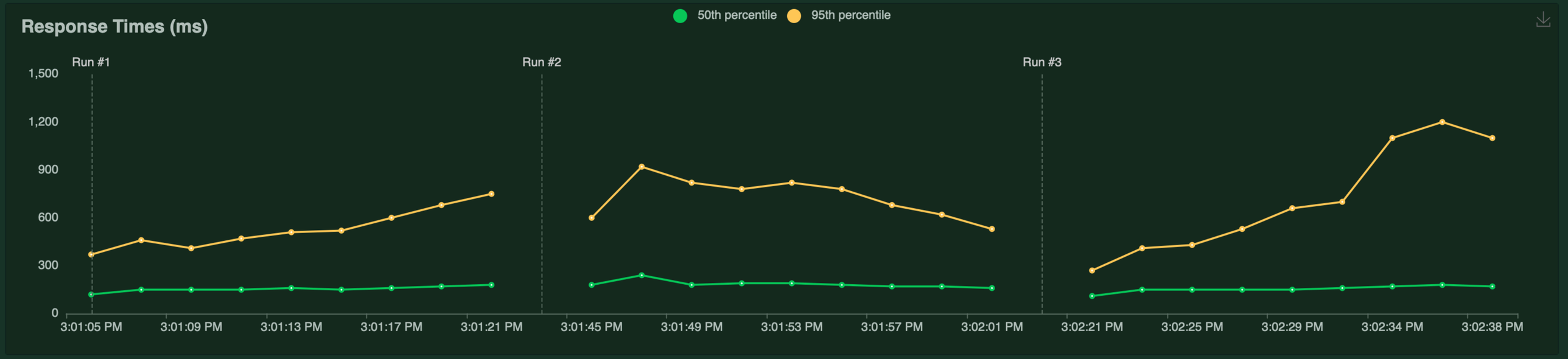
Read only app

db.sqlite3

```
COPY db.sqlite3 .
```

Readonly Benchmark





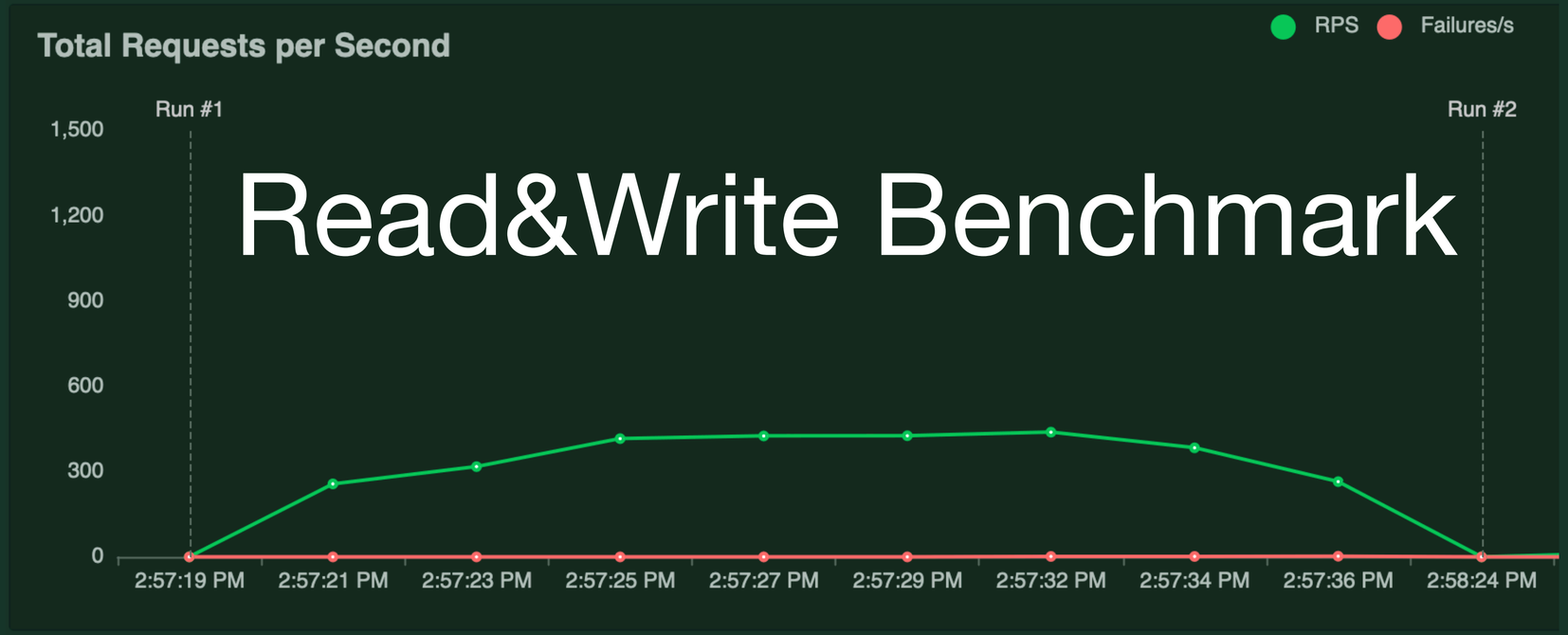
Honeymoon stage

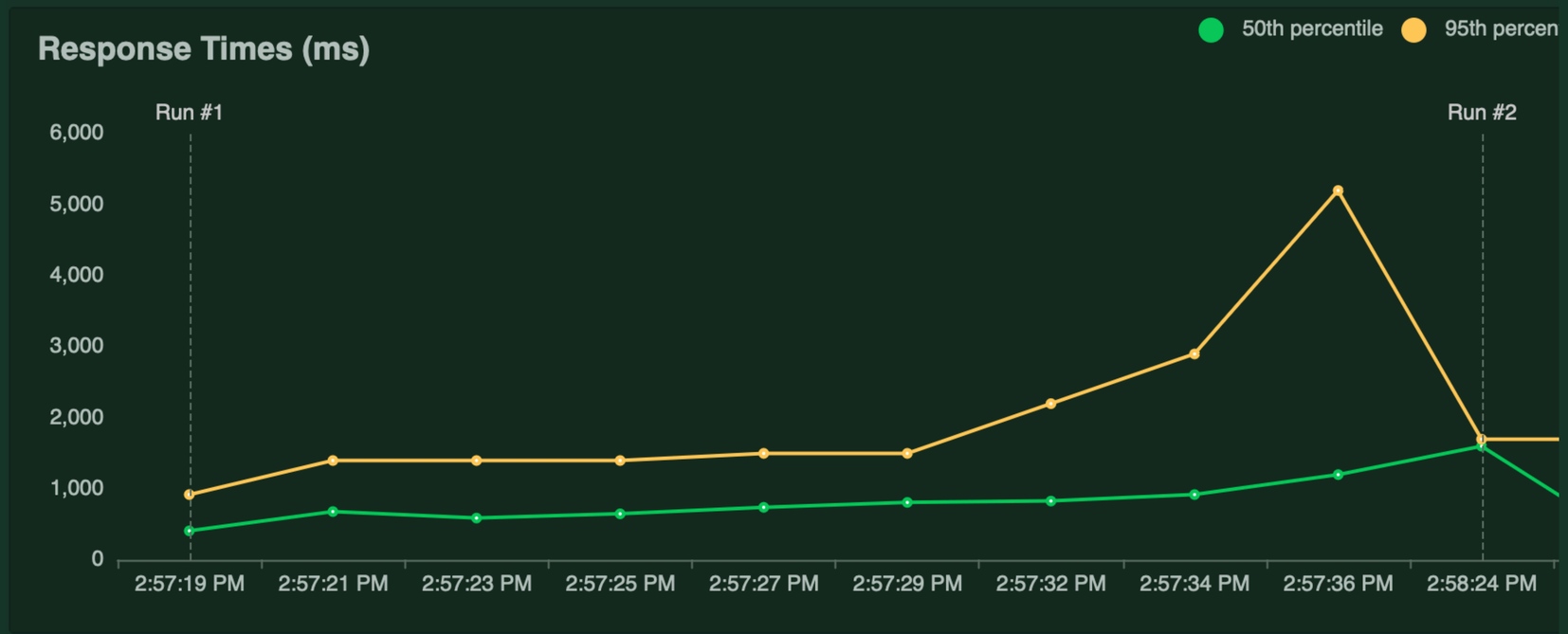
Writes

Docker 🛑

Writes block

Read&Write Benchmark

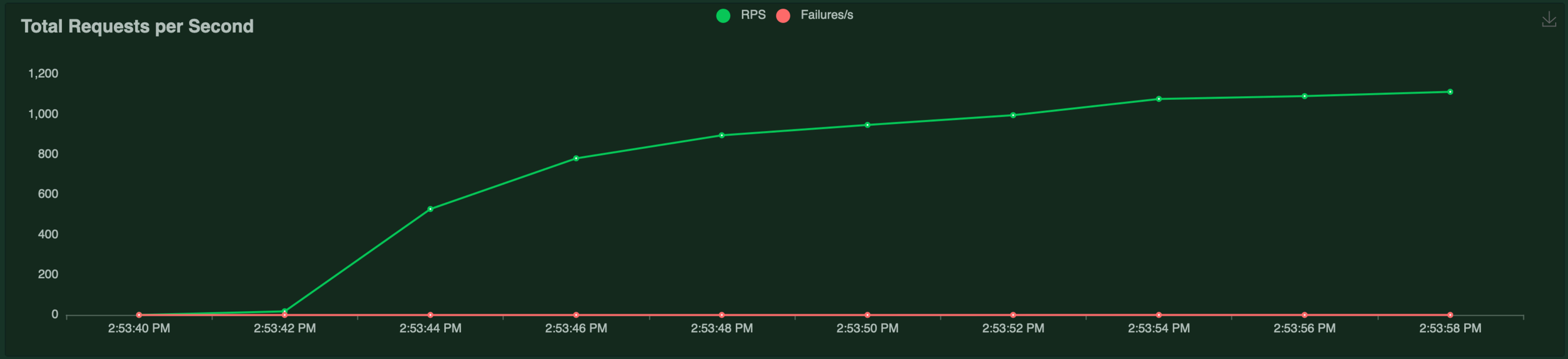


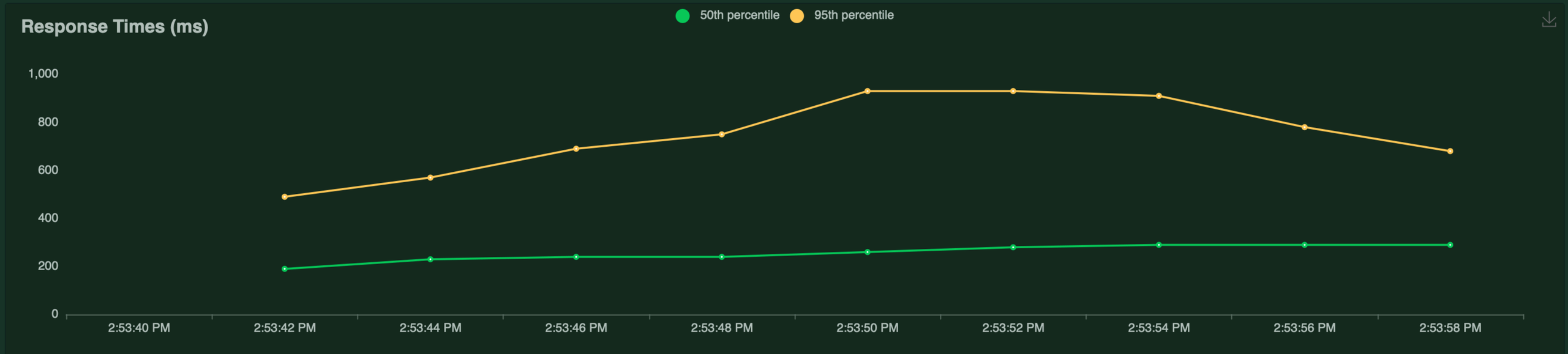


Unblock reads

```
PRAGMA journal_mode='WAL'
```

Read & Write Benchmark (WAL)





Database is locked

```
timeout=999999
```

New issue

We notified recently active members in the fedidevs project of this issue

ISSUE

OperationalError /

database is locked

Aug. 28, 2023, 6:38:35 a.m. UTC ID: 5d7b215636954b5488547295551d8d21

project	fedidevs
environment	production
level	error

Suspect Commits

Strip whitespace

c5a1103 — **Anže Pečar**

Exception

```
OperationalError: database is locked
  File "django/db/backends/utils.py", line 89, in _execute
    return self.cursor.execute(sql, params)
  File "django/db/backends/sqlite3/base.py", line 328, in execute
    return super().execute(query, params)

OperationalError: database is locked
(15 additional frame(s) were not displayed)
...
  File "accounts/views.py", line 41, in index
    page_obj = paginator.get_page(page_number)
```

Crisis stage

Transactions

Deferred transactions

```
BEGIN;
SELECT * FROM auth_user;
UPDATE auth_user SET last_login_date = NOW();
COMMIT;
```

Deferred transactions

```
BEGIN;
SELECT * FROM auth_user;
UPDATE auth_user SET last_login_date = NOW();
COMMIT;
```

Deferred transactions

```
BEGIN;
SELECT * FROM auth_user;
UPDATE auth_user SET last_login_date = NOW();
COMMIT;
```

Deferred transactions

```
BEGIN;
SELECT * FROM auth_user;
UPDATE auth_user SET last_login_date = NOW();
COMMIT;
```

Deferred transactions

```
BEGIN;
SELECT * FROM auth_user;
UPDATE auth_user SET last_login_date = NOW();
COMMIT;
```

Deferred transactions

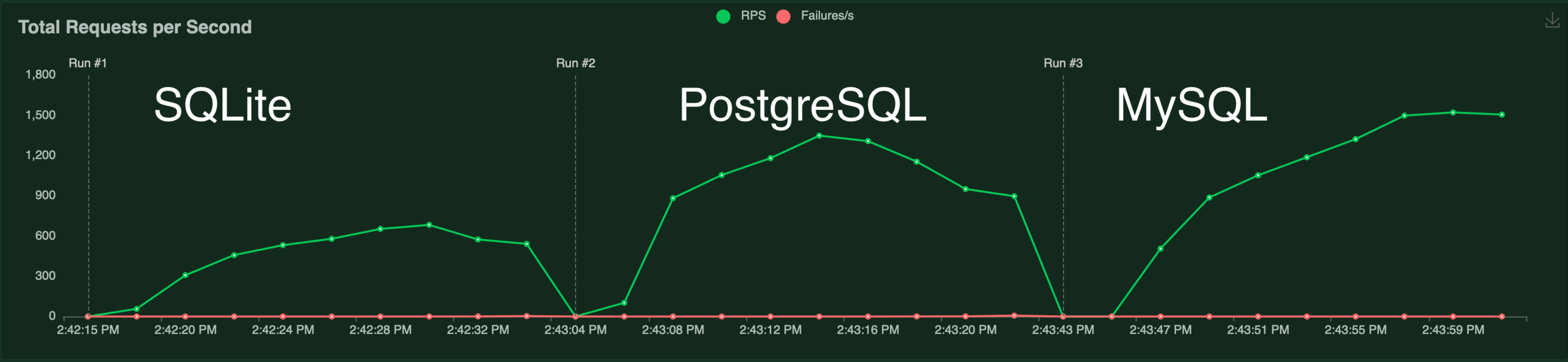
```
BEGIN;
SELECT * FROM auth_user;
UPDATE auth_user SET last_login_date = NOW();
COMMIT;
```

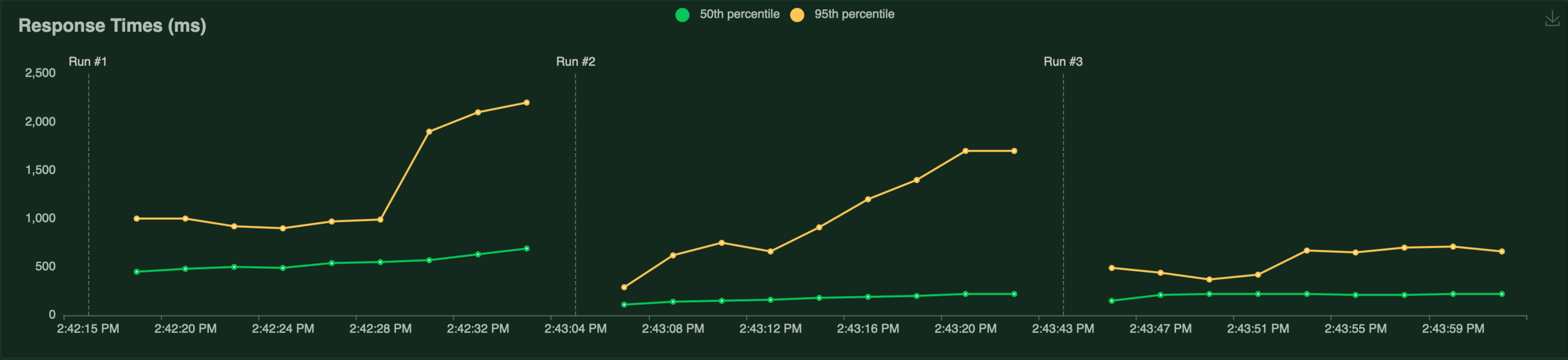
```
BEGIN IMMEDIATE;
SELECT * FROM auth_user;
UPDATE auth_user SET last_login_date = NOW();
COMMIT;
```

Write heavy

One concurrent write per database

Writeonly Benchmark





Multiple db.sqlite3 files

Beyond a single server

LiteFS

Backups

copy/paste is not safe

```
sqlite3 my.db ".backup 'my.db.bck'"
```

Litestream

Partnership stage

New in Django 5.1

```
DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.sqlite3",
        "NAME": BASE_DIR / "db.sqlite3",
        "OPTIONS": {
            "transaction_mode": "IMMEDIATE",
            "init_command": "PRAGMA journal_mode='WAL'", # <-- Enable WAL
        },
    }
}
```

Fixed #29280 -- Made the transactions behavior configurable on SQLite. #17760

Edit | Code

Merged **felixxm** merged 1 commit into `django:main` from `anze3db:sqlite_transaction_mode` on Jan 30

Conversation 34 | Commits 1 | Checks 18 | Files changed 5 | +112 −3

anze3db commented on Jan 19 • edited — Contributor

Opening this as a draft pull request until I figure out the following:

- [x] Where to add validation for the transaction_mode values?

Right now it's in `get_connection_params`, but it doesn't feel like the right place. We do validation for `NAME` there though, so maybe it's ok?

- [x] Is `OPTIONS` the right place for this `transaction_mode`?

Just double checking, since it doesn't feel clean that I have to pop it from the `OPTIONS` dict to avoid `TypeError: 'transaction_mode' is an invalid keyword argument for Connection()`

- [x] Figure out which documentation files to update

🎉 3

Reviewers

- charettes
- felixxm
- github-actions[bot]

Assignees

- felixxm

Labels

None yet

Projects

None yet

```
DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.sqlite3",
        "NAME": BASE_DIR / "db.sqlite3",
        "OPTIONS": {
            "transaction_mode": "IMMEDIATE",
            "init_command": "PRAGMA journal_mode='WAL'; ...",
        },
    }
}
```

```
PRAGMA journal_mode = WAL;
PRAGMA synchronous = NORMAL;
PRAGMA mmap_size = 134217728; -- 128 megabytes
PRAGMA journal_size_limit = 27103364; -- 64 megabytes
PRAGMA cache_size = 2000;
```

https://fractaledmind.github.io/2024/04/15/sqlite-on-rails-the-how-and-why-of-optimal-performance/

Future improvements?

Don't change defaults

startproject

Documentation

More SQLite topics!

- Litestream/LiteFS
- Full text search
- Cache
- Pub Sub
- Job Processor Backend (django-task?)

Questions?

https://pecar.me

@anze3db@fosstodon.org

Questions?

https://pecar.me

@anze3db@fosstodon.org

Follow for more relationship advice

ACID Isolation

READ UNCOMMITED

READ COMMITED

REPEATABLE READ

SERIALIZABLE

READ UNCOMMITED*

READ COMMITED

REPEATABLE READ

SERIALIZABLE

READ UNCOMMITED

READ COMMITED

REPEATABLE READ

SERIALIZABLE

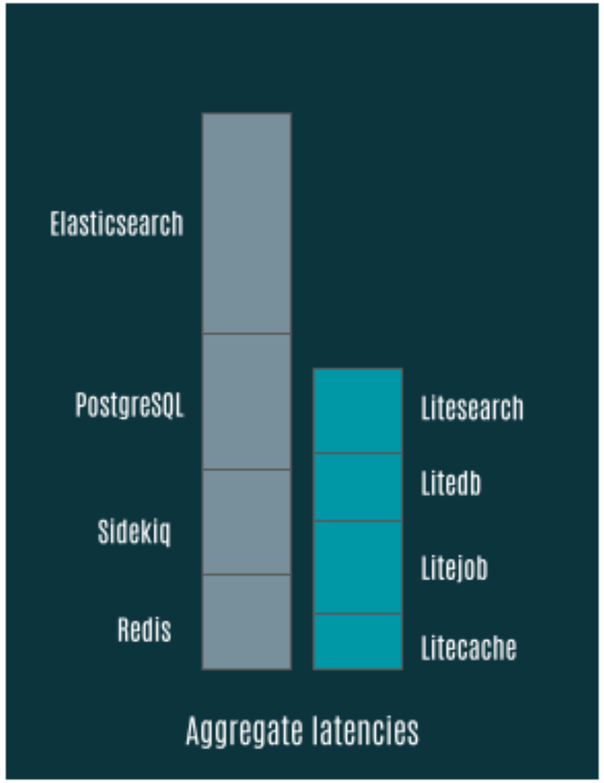
Litestack

```
# database connection
gem “pg”

# cache, cable & queue
gem “redis”
gem “hiredis”

# job processing
gem “sidekiq”

# full text search
gem “elasticsearch-rails”

# performance monitoring
gem “rails_performance”
```

Turn this ..

```
# almost everything
gem “litestack”
```

.. into this ..



.. and get this!