SANIC

EXPERIENCE BLAZING FAST PYTHON WEB FRAMEWORK

PyCon ID Surabaya, Indonesia 2017

Fauzan Erich Emmerling

VP Integration @ prismapp.io

@femmerling (GitHub, twitter, IG)

pythonista since 2010

---

http://www.emfeld.com

currently active in Python-ID Jogja

THEY SAY

PYTHON WEB FRAMEWORKS ARE SLOW

WHY WAS PYTHON SLOW

- It’s interpreted
- It’s dynamically typed
- In python 2.x concurrency is a big mess
- Async is hard
- The notoriously famous GIL

THEN COMES PYTHON 3.X

WHAT HAPPENS AT 3.X??

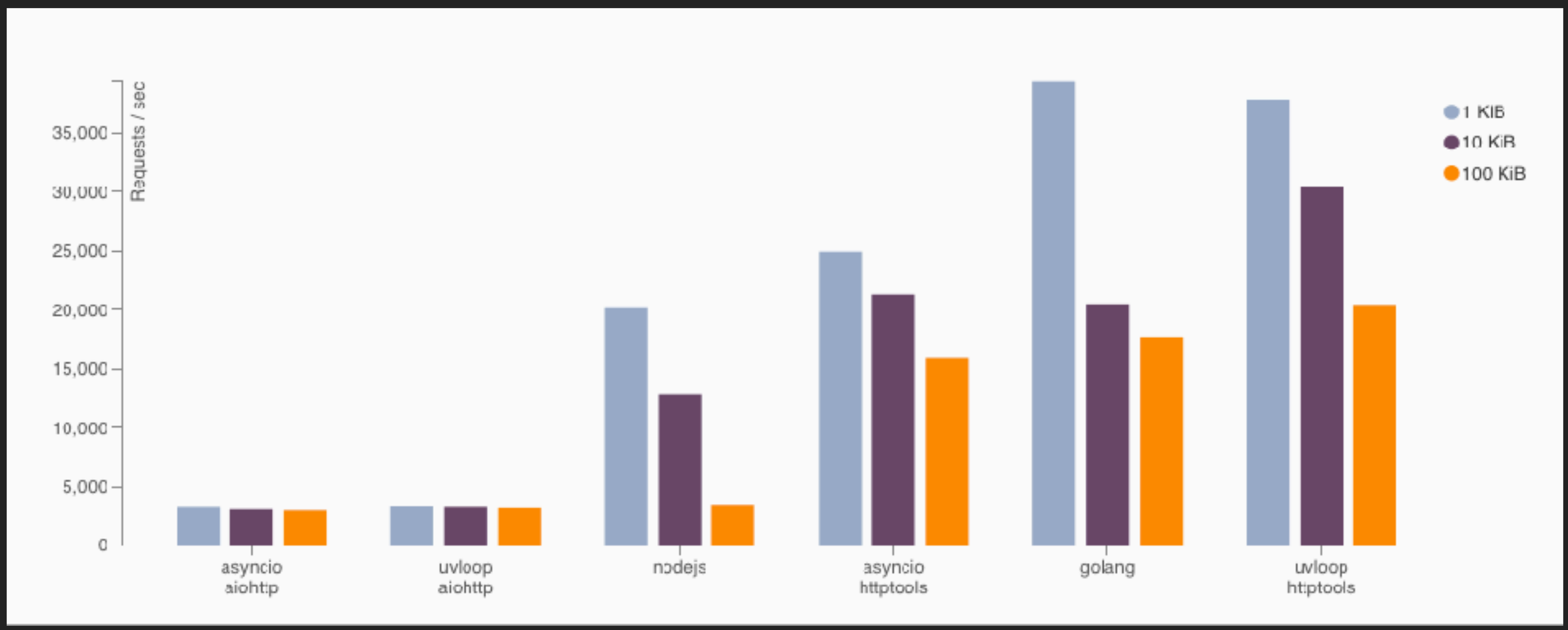
- Still interpreted
- Better performance
- Still dynamically typed but now comes with type hinting
- Since 3.5 async is a walk in the park
- GIL is still there but Asyncio saves us, BIG TIME

ENTER UVLOOP

"a full, drop-in replacement for the asyncio event loop. uvloop is written in Cython and built on top of libuv.

uvloop makes asyncio fast. In fact, it is at least **2x faster** than nodejs, gevent, as well as any other Python asynchronous framework"

Yuri Selivanov



HTTP Benchmark Test

GOTTA GO FAST!!!

DUDE, IT'S MY TURN NOW!!!

SANIC WEB FRAMEWORK

WHAT IS IT?

- a python 3.5+ micro framework thats written to go fast
- small and contained
- inspired a lot by Flask, that's why I ❤️ it
- accepts async handlers so the we can use shiny async/await syntax

WHY USE SANIC?

- super easy to learn
- can be a micro framework up to a full monolith, just like flask
- async capabilities, better concurrency
- performs like that of Go or NodeJS but with better syntax
- great docs
- extensions are growing
- however currently no wheel for uvloop on windows yet
- set your own workers

FEATURES

- async request handling
- http and websocket
- class based views
- blueprints
- configurations
- cookies

HOW TO START USING IT?

- make sure you have python 3.5+
- $ pip install sanic
- start coding

DOES THIS RING A BELL?

```
from sanic import Sanic
from sanic import response as sanic_response
from services import get_image_content, get_mimetype, set_expiry_headers

app = Sanic(__name__)

@app.route('/<image_url:path>')
async def index(request, image_url):
    content_type = get_mimetype(image_url=image_url)
    headers = set_expiry_headers()
    async def get_image(response):
        content = await get_image_content(image_url)
        response.write(content)
    return sanic_response.stream(
            get_image,
            content_type=content_type,
            headers=headers)

if __name__ == '__main__':
    app.run(host='0.0.0.0',
        port=5000, debug=True, workers=4)
```

SOME BEST PRACTICES

use factory pattern for apps:

```
def create_app():
    app = Sanic(__name__)

    init_exception_handling(app=app)


app.add_route(
    handle_transcript,
    '/v2/conversations_transcript',
    methods=['POST'])

app.add_route(
    handle_ack,
    '/v2/conversations_transcript',
    methods=['OPTIONS'])

return app
```

SOME BEST PRACTICES

map exceptions

```
def init_handled_exceptions(app):
    if app is None:
        raise AppError('`app` is required')
    @app.exception(AppError)
def handled_exceptions(request, exception):
    error = create_response(
        payload=exception.to_dict(),
        status_code=exception.status_code)
    logger.error(
        exception.message,
        error=error)
    return response.json(
        error.payload,
        headers=error.headers,
        status=exception.status_code)
```

SOME BEST PRACTICES

separate handlers from route for easier testing

```
from app.handlers.transcript import handle_transcript

app = Sanic(__name__)

    init_exception_handling(app=app)

    app.add_route(
        handle_transcript,
        '/v2/conversations_transcript',
        methods=['POST'])
```

SANIC IN REAL LIFE

- Scarlett Project
- Translates image streams from http to https
- Handles around 200 req/s per worker
- Runs behind Gunicorn with 4 workers running
- Only need 1 docker container for now
- Average response time is 10ms
- Developed in 1.5 hours

SOMETIMES YOU GOTTA RUN BEFORE YOU CAN WALK

Tony Stark

QUESTIONS?

you can also ask on:

twitter: @femmerling

mail: erich@emfeld.com

THANK YOU

in case you’re wondering..

Yes, we’re hiring!!!

fauzan@prismapp.io