

Рубрика «Ээксперименты!»

Цели:

- Понять преимущества асинхронного подхода перед синхронным
- Понять последствия использования синхронных операций в асинхронном коде

Задание 7. Сравнение производительности aiohttp и Django при увеличении количества параллельных запросов

Запустите команды ab с соответствующими параметрами и заполните таблицу результатов в отчете. Изменяемый параметр: **число параллельных соединений**, значения: **5, 10, 20, 40**.

```
ab -n 100 -c <Число параллельных соединений> http://127.0.0.1:8088/
```

```
ab -n 100 -c <Число параллельных соединений> http://127.0.0.1:8089/
```

Постройте графики для aiohttp и Django и приложите их к отчету. Графики:

1. ось X — количество параллельных запросов, ось Y — время запроса. На графике должно быть 2 линии: per request, all concurrent
2. ось X — количество параллельных запросов, ось Y — количество RPS

Задание 8. Сравнение производительности aiohttp и Django при увеличении времени io bound операции

Одновременно изменяйте параметр sleep в aiohttp и Django. Изменяемый параметр: **время сна**, значения: **0.1, 0.2, 0.3**.

PS: не забывайте перезапускать веб-серверы, иначе параметры не будут меняться.

```
# servers/aio/server.py
await asyncio.sleep(0.1)

# servers/dj/dj/urls.py
time.sleep(0.1)
```

Запустите команды ab с соответствующими параметрами и заполните таблицу результатов в отчете.

```
ab -n 100 -c 10 http://127.0.0.1:8088/
```

```
ab -n 100 -c 10 http://127.0.0.1:8089/
```

Постройте графики для aiohttp и Django и приложите их к отчету. Графики:

1. ось X — время сна, ось Y — время запроса. На графике должно быть 2 линии: per request, all concurrent
2. ось X — время сна, ось Y — количество RPS

Задание 9. Сравнение производительности aiohttp и Django при увеличении количества параллельных запросов, если в aiohttp использовать синхронную операцию.

```
# замените в файле servers/aio/server.py
# строку
await asyncio.sleep(0.1)
# на
time.sleep(0.1)
# и перезапустите aiohttp
```

Запустите команды ab с соответствующими параметрами и заполните таблицу результатов в отчете. Изменяемый параметр — **число параллельных соединений**, значения: **5, 10, 20, 40**.

```
ab -n 100 -c <Число параллельных соединений> http://127.0.0.1:8088/
```

Постройте графики для aiohttp, сравните их с Django из задания 7 и приложите к отчету. Графики:

1. ось X — количество параллельных запросов, ось Y — время запроса. На графике должно быть 2 линии: per request, all concurrent
2. ось X — количество параллельных запросов, ось Y — количество RPS

Что нужно понять, когда проведете ээксперименты:

- Как ведет себя aiohttp при увеличении количества одновременных клиентов
- Как ведет себя Django при увеличении количества одновременных клиентов
- Как ведет себя aiohttp при увеличении времени выполнения io-операции
- Как ведет себя Django при увеличении времени выполнения io-операции
- Как ведет себя aiohttp при использовании синхронной операции