

วิธีการ Benchmark API

การ benchmark API คือการทดสอบประสิทธิภาพของ API เพื่อวัดความเร็ว ความสามารถในการรองรับจำนวนคำขอ (concurrency) และการใช้ทรัพยากรระบบ วิธีการทำมีขั้นตอนหลัก ๆ ดังนี้:

1. เตรียม API ที่จะทดสอบ

- สร้าง API endpoint ที่ต้องการทดสอบ เช่น route ที่ตอบกลับข้อความ "Hello, World!" หรือ API จริงที่ให้บริการข้อมูล
- ตรวจสอบให้ API ทำงานได้ปกติและพร้อมรับคำขอจำนวนมาก

2. เลือกเครื่องมือ Benchmark

เครื่องมือยอดนิยมสำหรับ benchmark API เช่น:

- **wrk**: เครื่องมือบรรทัดคำสั่งสำหรับทดสอบโหลด HTTP ที่รองรับการตั้งค่าคอนเคอร์เรนซี (concurrency), จำนวนเธรด, ระยะเวลาทดสอบ และแสดงผล latency, requests per second เป็นต้น
- **rewrk**: เครื่องมือ benchmark สำหรับ Rust web frameworks ที่ใช้งานง่าย
- **Apache Benchmark (ab)**: เครื่องมือพื้นฐานสำหรับทดสอบ HTTP load
- เครื่องมืออื่น ๆ เช่น JMeter, k6, Gatling (สำหรับการทดสอบขั้นสูง)

3. ตั้งค่าการทดสอบ

- กำหนดจำนวน concurrent connections (เช่น 100-1000)
- กำหนดจำนวนเธรด (threads) ที่ใช้
- กำหนดระยะเวลาทดสอบ (เช่น 8-10 วินาที)
- ตัวอย่างคำสั่ง wrk:

```
wrk --latency -t4 -c200 -d8s http://127.0.0.1:8080/api/endpoint
```

โดย

- `-t4` = ใช้ 4 threads
- `-c200` = เปิด 200 concurrent connections
- `-d8s` = รันทดสอบ 8 วินาที
- `--latency` = แสดงข้อมูล latency เพิ่มเติม

4. รันการทดสอบและเก็บข้อมูล

- รันคำสั่ง benchmark หลายรอบเพื่อให้ได้ข้อมูลที่น่าเชื่อถือ
- เก็บข้อมูลสำคัญ เช่น Requests per second, Latency (average, max, min), Error rate, Throughput (transfer rate)

5. วิเคราะห์ผล

- เปรียบเทียบ requests per second เพื่อดูว่า API รับโหลดได้เท่าไร
- ดู latency เพื่อประเมินความเร็วตอบสนอง
- ตรวจสอบ error rate ว่ามีค่าขอที่ล้มเหลวหรือไม่
- ประเมินการใช้ทรัพยากร เช่น CPU, RAM ระหว่างทดสอบ

6. ปรับแต่งและทดสอบซ้ำ

- ปรับปรุงโค้ดหรือการตั้งค่าของ API (เช่นเพิ่ม caching, optimize query)
- ทดสอบซ้ำเพื่อดูผลลัพธ์ที่ดีขึ้น

ตัวอย่างการ Benchmark Rust Web Frameworks

- ในการเปรียบเทียบ Actix-web, Warp, Axum, Rocket ใช้ `wik` เป็นเครื่องมือ benchmark โดยตั้งค่าคอนเคอร์เรนซี 200, 4 threads, 8 วินาที
- ตัวอย่างคำสั่ง:

```
wik --latency -t4 -c200 -d8s http://127.0.0.1:8081
```

- ผลลัพธ์จะบอกจำนวน requests per second, latency, และ throughput เพื่อเปรียบเทียบประสิทธิภาพแต่ละ framework ^[1] ^[2].

สรุป

- ใช้เครื่องมือ benchmark เช่น `wik` หรือ `rewik`
- ตั้งค่าจำนวน concurrent connections, threads, ระยะเวลาทดสอบให้เหมาะสม
- รันทดสอบหลายรอบเพื่อความแม่นยำ
- วิเคราะห์ผล requests/sec, latency, error rate, throughput
- ปรับปรุง API และทดสอบซ้ำเพื่อเพิ่มประสิทธิภาพ

การ benchmark อย่างถูกวิธีช่วยให้เข้าใจขีดจำกัดของ API และเลือกเทคโนโลยีหรือการปรับแต่งที่เหมาะสมได้ดียิ่งขึ้น ^[1] ^[3] ^[4].

✱

1. <https://github.com/rousan/rust-web-frameworks-benchmark>

2. <https://github.com/piaoger/webframework-bench>

3. <https://www.youtube.com/watch?v=MY3Ou5ehwUQ>

4. <https://www.youtube.com/watch?v=Tggh0v7nBzs>