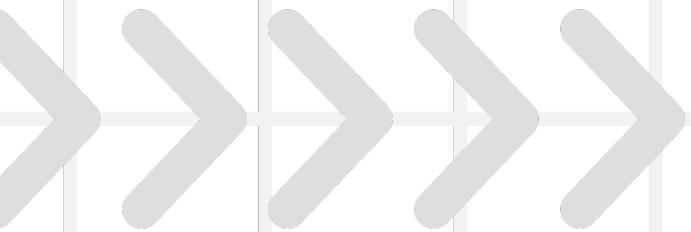


LOCUST

Nhóm C2 - Bulb Hacking Team



Performance Testing

Kiểm thử phần mềm | 2023

THÔNG TIN NHÓM

MSSV	Họ và tên	Nhóm trưởng
20120406	Phạm Quốc Vương	Nhóm trưởng
20120251	Trần Đức Anh	
20120385	Trần Hoàng Tín	
20120435	Lê Thị Ngọc Bích	



PERFORMANCE SOFTWARE

PERFORMANCE SOFTWARE

1. Application Performance

Được hiểu ngắn gọn là hiệu suất hoạt động của ứng dụng.

Cho phép người dùng sử dụng, thực hiện các tính năng mà không cảm thấy quá chậm trễ hay khó chịu.

Tùy thuộc vào quan điểm và cảm nhận của người dùng

PERFORMANCE SOFTWARE

2. Các chỉ số để đo performance

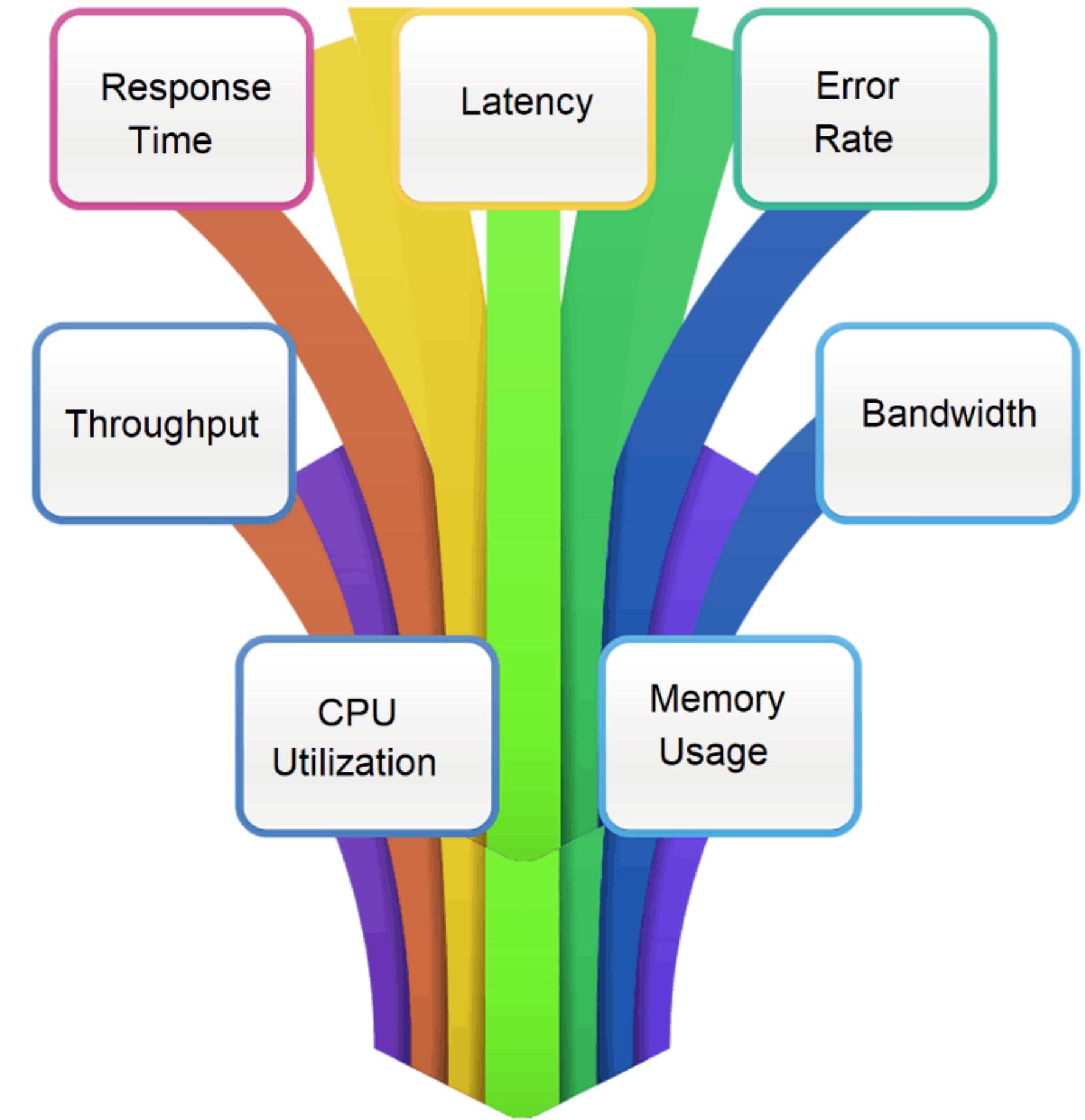
Service-oriented: hướng đến khả năng cung cấp dịch vụ cho khách hàng

- **Availability:** Tính khả dụng để người dùng bắt đầu sử dụng.
- **Response Time:** Thời gian phản hồi kết quả đến người dùng.

Efficiency-oriented: Hướng đến hiệu năng của phần mềm

- **Throughput:** Tỉ lệ số lượng request mà ứng dụng có thể xử lý trên đơn vị thời gian (theo giờ, phút, giây,...).
- **Utilization:** Tỷ lệ dung lượng của một tài nguyên đang được sử dụng

Key Performance Indicators



PERFORMANCE SOFTWARE

3. Performance kém

NGUYÊN NHÂN

- Sử dụng công nghệ cũ, phụ thuộc và ràng buộc với các phiên bản làm khó đạt hiệu năng cao.
- Các vấn đề về tính tương thích trong hệ thống.
- Thiết kế ứng dụng không cân nhắc về hiệu suất.
- Ứng dụng phát triển hay mở rộng vượt ngoài dự tính.
- Triển khai thực tế khác biệt quá lớn so với quá trình phát triển sản phẩm.
- Không bảo trì ứng dụng sau khi triển khai thực tế.

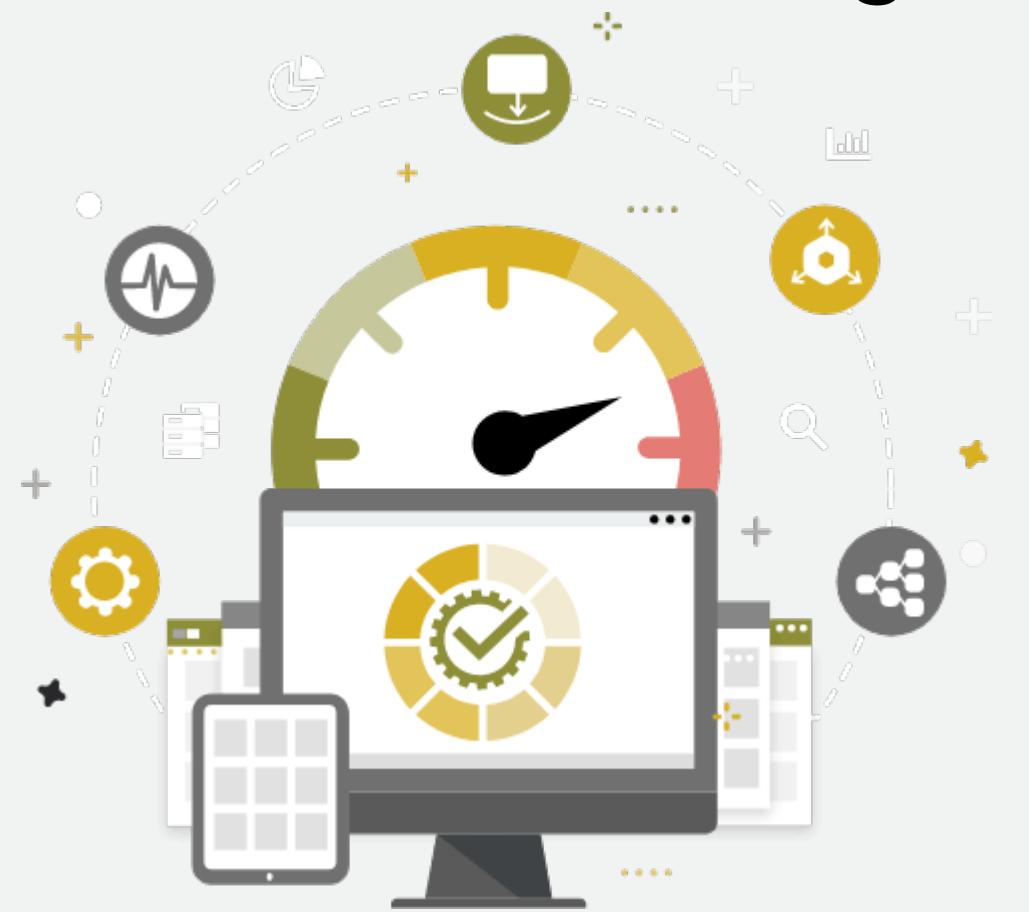
HẬU QUẢ

- Giảm năng suất.
- Mất khách hàng.
- Mất uy tín.
- Tiêu tốn nguồn lực để sửa chữa, cải thiện.

PERFORMANCE TESTING

PERFORMANCE TESTING

1. Performance testing là gì?



Kiểm tra các khía cạnh của một phần mềm:

- Tốc độ
- Thời gian phản hồi
- Độ tin cậy
- Mức độ sử dụng tài nguyên
- Lượng người dùng tối đa.
- Xác định và loại bỏ các vấn đề tắc nghẽn hiệu xuất, đo lường khả năng chịu tải và khả năng duy trì hiệu suất mong muốn.

PERFORMANCE TESTING

2. Vì sao cần Performance testing.

Đảm bảo hiệu suất.

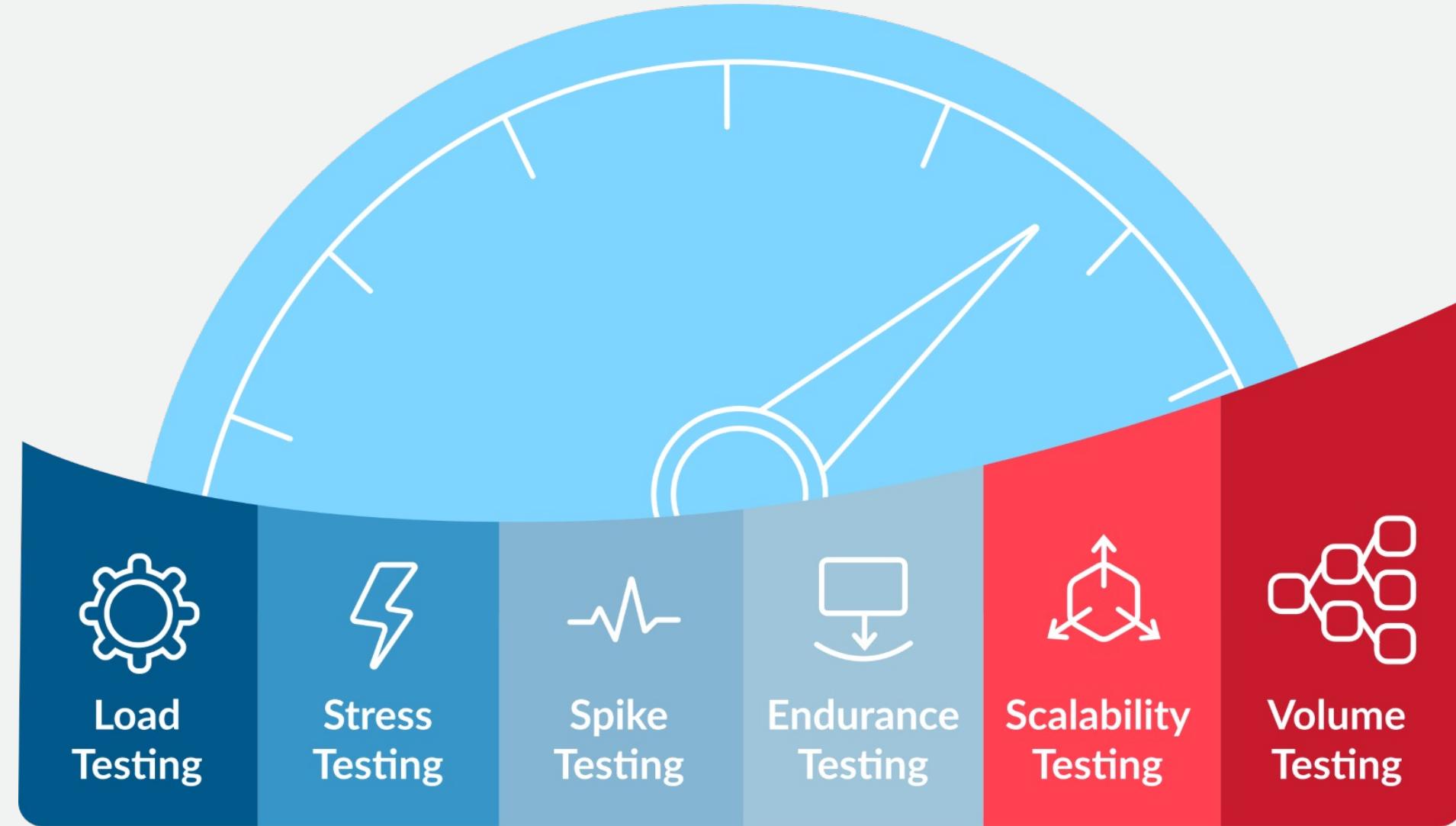
Phát hiện vấn đề tiềm ẩn.

Giúp đề ra chiến lược tối ưu hóa tài nguyên tương ứng.

Đáp ứng Business requirements.

PERFORMANCE TESTING

3. Phân loại



PERFORMANCE TESTING

3. Phân loại

Baseline

Xác định ngưỡng chịu tải tối đa của hệ thống

Load

Testing

Throughput/lượng user: Cao (vượt ngưỡng thông thường)

Thời gian test: Trung bình (5-60 phút)

Mục đích: Kiểm tra khả năng chịu tải

Ví dụ: Ngày trả lương, giờ cao điểm, ngày cuối tuần,...

Stress

Testing



Stress test

VUs/Throughput

Above
Average

Average

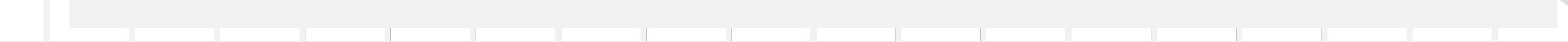
Few

Short

Medium

Very Long!

Duration



Soak test

VUs/Throughput

Above
Average

Average

Few

Short

Medium

Very Long!

Duration



PERFORMANCE TESTING

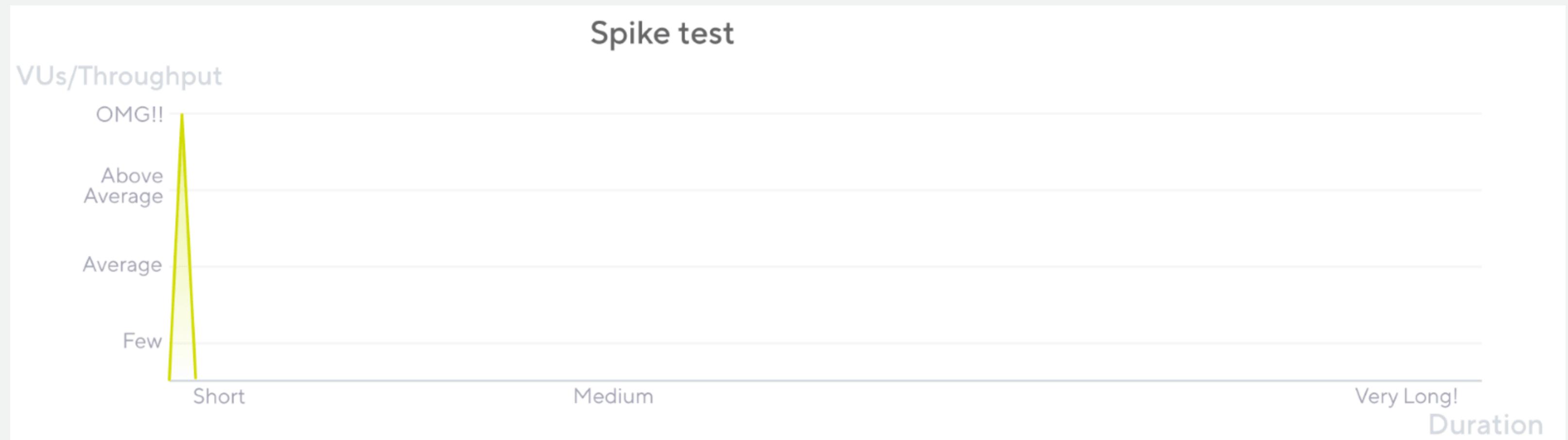
3. Phân loại

Endurance (Soak) Testing

Throughput/lượng user: Trung bình, như bình thường
Thời gian test: Dài (hàng tiếng đồng hồ)
Mục đích: Kiểm tra khả năng sử dụng dài hạn.
Ví dụ: Ứng dụng hoạt động giờ hành chính mở rộng sang 24/7

Spike Testing

Throughput/lượng user: Rất cao
Thời gian test: đủ dài để kiểm tra cách hệ thống xử lý (vài phút)
Mục đích: Đảm bảo hệ thống sẵn sàng cho các đợt tấn công mạng hay các event ngắn hạn.
Ví dụ: Săn sale shopee, Đăng ký học phần.



PERFORMANCE TESTING

3. Phân loại

Volume Testing

Kiểm thử phi chức năng

Kiểm tra khả năng chịu tải của phần mềm với một lượng dữ liệu nhất định

Thay đổi kích thước cơ sở dữ liệu để kiểm tra hiệu năng của phần mềm.

Scalability Testing

Xác định tính hiệu quả của phần mềm trong việc nhân rộng (ví dụ như tăng lượng người dùng)

Hỗ trợ tăng tải cho người dùng

Giúp lập kế hoạch bổ sung khả năng đáp ứng của hệ thống với nhu cầu tăng dần.

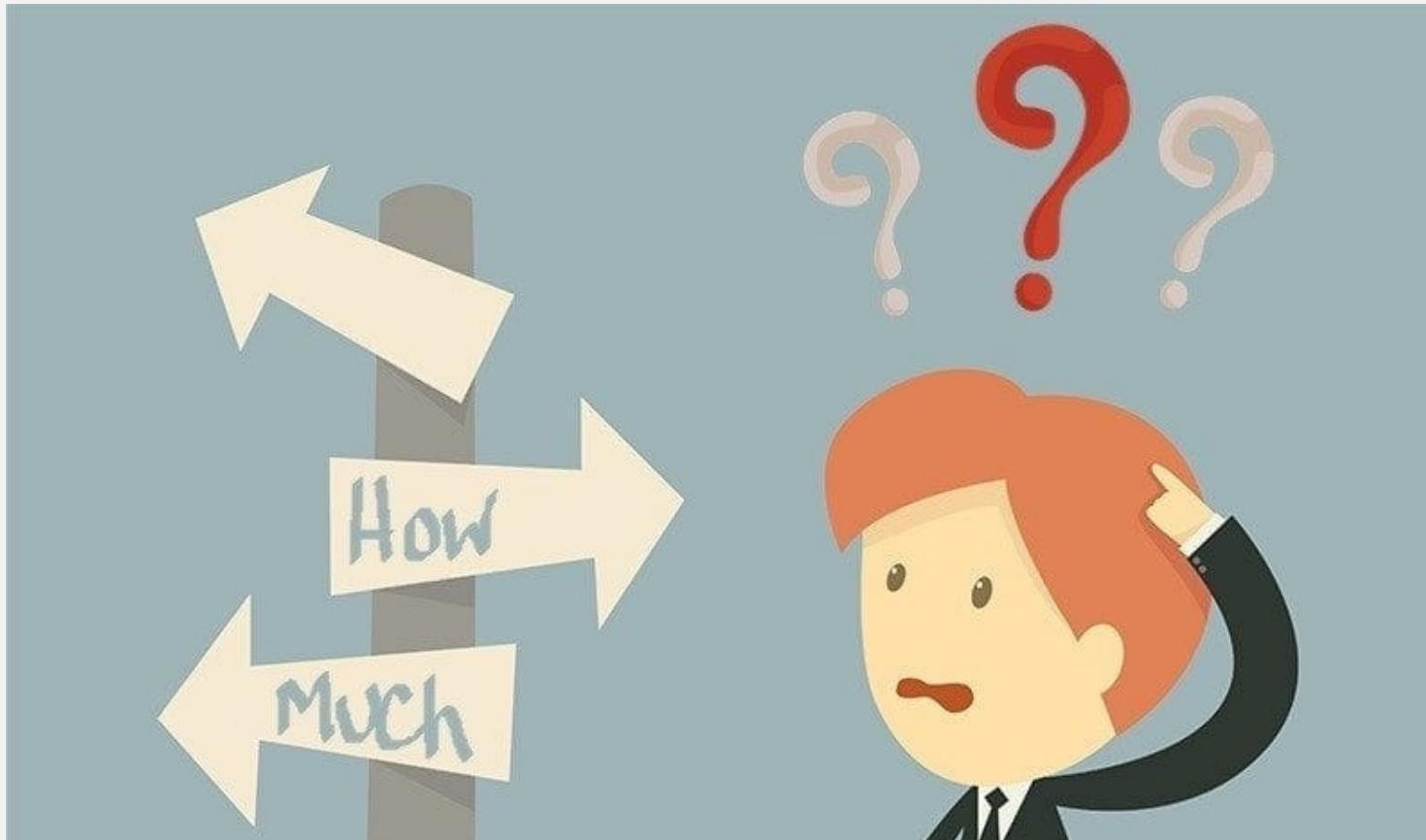
PERFORMANCE TESTING

4. Quy trình Performance Testing

- Xác định yêu cầu
- Lập kế hoạch
- Thiết kế test
- Thiết lập môi trường
- Thực hiện kiểm thử
- Phân tích và báo cáo kết quả

PERFORMANCE TESTING

5. Chi phí thực hiện performance testing



- Phần cứng
- Công cụ
- Giấy phép
- Nhân lực
- Thời gian
- Môi trường kiểm thử
- Chi phí liên quan

PERFORMANCE TESTING

5. Công ty nổi tiếng trong lĩnh vực Performance Testing

Neotys
(Neoload)



Micro Focus
(LoadRunner)



Apache
(Apache Jmeter)



BlazeMeter



RadView Sofware
(WebLOAD)





LOCUST





LOCUST

1. Về công cụ

Tác giả: Jonatan Heyman, Carl Byström, Joakim Hamréen, Hugo Heyman, ...

Giấy phép: Giấy phép mã nguồn mở MIT.

Cơ sở phát triển:

- Xuất phát từ những hạn chế của các giải pháp performance testing.
- Công cụ lúc đó có giao diện phức tạp, cấu hình rườm rà.
- Tạo ra Locust, cho phép tester mô phỏng hoạt động của người dùng bằng code python đơn giản.

Tính năng:

- Viết script test bằng python.
- Hỗ trợ giả lập hàng ngàn người dùng logic đồng thời.
- Web-based UI
- Có thể test bất cứ hệ thống nào



LOCUST

HOST
<https://www.fahasa.com>

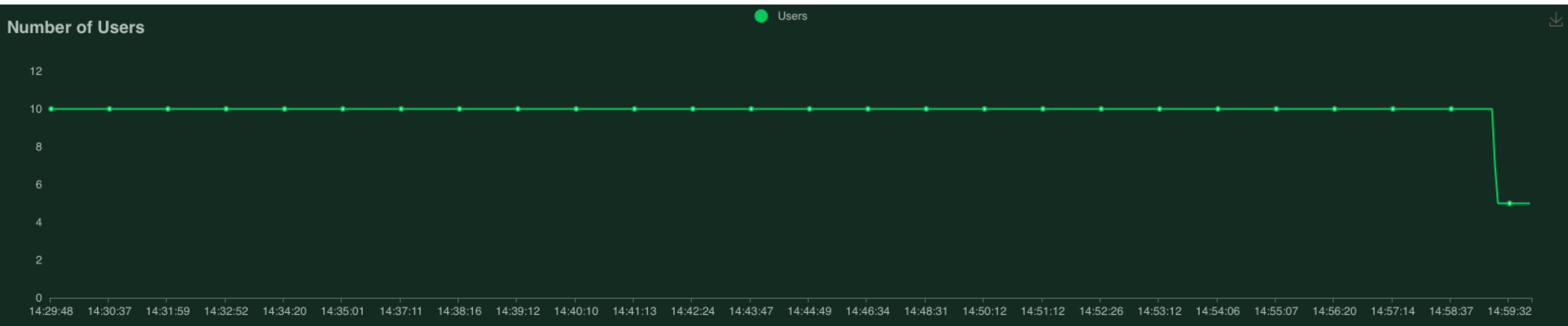
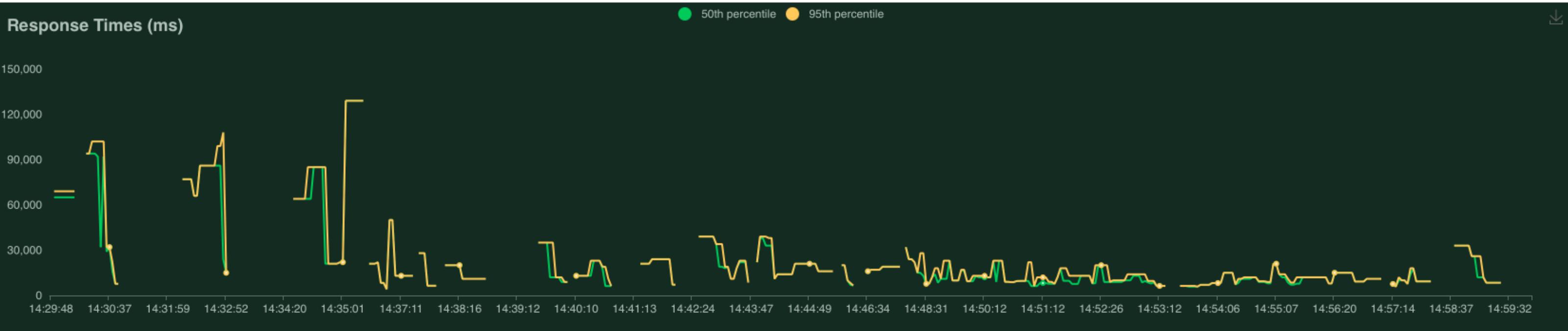
STATUS
STOPPED
New test

RPS
2.7

FAILURES
0%

[Statistics](#) [Charts](#) [Failures](#) [Exceptions](#) [Current ratio](#) [Download Data](#)

Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
POST	/checkout_api/cart/add	20	0	34	41	42	34	28	42	1325	1.3	0
GET	/tet-o-lang-dia-nguc.html	22	0	220	290	380	239	143	379	1106230	1.4	0
	Aggregated	42	0	140	270	380	142	28	379	580085	2.7	0



- ret1na@ret1na-G3-3500:~/20120251/KiemThuPhanMem/Locust-Seminar/Features Demo\$ locust -f feature_test.py

```
[2023-11-06 13:48:59,811] ret1na-G3-3500/INFO/locust.main: Starting web interface at http://0.0.0.0:8089 (accepting connections from all network interfaces)
[2023-11-06 13:48:59,817] ret1na-G3-3500/INFO/locust.main: Starting Locust 2.17.0
[2023-11-06 13:49:04,931] ret1na-G3-3500/INFO/locust.runners: Shape test starting.
[2023-11-06 13:49:04,931] ret1na-G3-3500/INFO/locust.runners: Shape worker starting
[2023-11-06 13:49:04,931] ret1na-G3-3500/INFO/locust.runners: Shape test updating to 200 users at 10.00 spawn rate
Beginning test
[2023-11-06 13:49:04,932] ret1na-G3-3500/INFO/locust.runners: Ramping to 200 users at a rate of 10.00 per second
[2023-11-06 13:49:23,945] ret1na-G3-3500/INFO/locust.runners: All users spawned: {"WebsiteUser": 200} (200 total users)
Spawned ... 200 users.
[2023-11-06 13:50:44,989] ret1na-G3-3500/INFO/locust.runners: Shape test updating to 1 users at 10.00 spawn rate
[2023-11-06 13:50:44,989] ret1na-G3-3500/INFO/locust.runners: Ramping to 1 users at a rate of 10.00 per second
[2023-11-06 13:51:03,999] ret1na-G3-3500/INFO/locust.runners: All users spawned: {"WebsiteUser": 1} (1 total users)
Spawned ... 1 users.
[2023-11-06 13:51:15,005] ret1na-G3-3500/INFO/locust.runners: Shape test updating to 400 users at 15.00 spawn rate
[2023-11-06 13:51:15,005] ret1na-G3-3500/INFO/locust.runners: Ramping to 400 users at a rate of 15.00 per second
[2023-11-06 13:51:19,850] ret1na-G3-3500/WARNING/root: CPU usage above 90%! This may constrain your throughput and may even give inconsistent response time measurements! See https://docs.locust.io/en/stable/running-distributed.html for how to distribute the load over multiple CPU cores or machines
[2023-11-06 13:51:42,518] ret1na-G3-3500/INFO/locust.runners: All users spawned: {"WebsiteUser": 400} (400 total users)
Spawned ... 400 users.
[2023-11-06 13:54:05,718] ret1na-G3-3500/INFO/locust.runners: Shape test updating to 1 users at 15.00 spawn rate
[2023-11-06 13:54:05,718] ret1na-G3-3500/INFO/locust.runners: Ramping to 1 users at a rate of 15.00 per second
[2023-11-06 13:54:31,735] ret1na-G3-3500/INFO/locust.runners: All users spawned: {"WebsiteUser": 1} (1 total users)
Spawned ... 1 users.
[2023-11-06 13:54:35,736] ret1na-G3-3500/INFO/locust.runners: Shape test updating to 200 users at 10.00 spawn rate
[2023-11-06 13:54:35,736] ret1na-G3-3500/INFO/locust.runners: Ramping to 200 users at a rate of 10.00 per second
[2023-11-06 13:54:54,748] ret1na-G3-3500/INFO/locust.runners: All users spawned: {"WebsiteUser": 200} (200 total users)
Spawned ... 200 users.
[2023-11-06 13:59:05,913] ret1na-G3-3500/INFO/locust.runners: Shape test updating to 1 users at 10.00 spawn rate
[2023-11-06 13:59:05,913] ret1na-G3-3500/INFO/locust.runners: Ramping to 1 users at a rate of 10.00 per second
[2023-11-06 13:59:24,924] ret1na-G3-3500/INFO/locust.runners: All users spawned: {"WebsiteUser": 1} (1 total users)
Spawned ... 1 users.
[2023-11-06 14:00:04,953] ret1na-G3-3500/INFO/locust.runners: Shape test updating to 600 users at 60.00 spawn rate
[2023-11-06 14:00:04,953] ret1na-G3-3500/INFO/locust.runners: Ramping to 600 users at a rate of 60.00 per second
[2023-11-06 14:00:23,463] ret1na-G3-3500/INFO/locust.runners: All users spawned: {"WebsiteUser": 600} (600 total users)
Spawned ... 600 users.
[2023-11-06 14:00:45,633] ret1na-G3-3500/INFO/locust.runners: Shape test updating to 1 users at 60.00 spawn rate
[2023-11-06 14:00:45,634] ret1na-G3-3500/INFO/locust.runners: Ramping to 1 users at a rate of 60.00 per second
[2023-11-06 14:00:54,651] ret1na-G3-3500/INFO/locust.runners: All users spawned: {"WebsiteUser": 1} (1 total users)
Spawned ... 1 users.
[2023-11-06 14:01:15,651] ret1na-G3-3500/INFO/locust.runners: Shape test stopping
[2023-11-06 14:01:15,651] ret1na-G3-3500/WARNING/locust.runners: CPU usage was too high at some point during the test! See https://docs.locust.io/en/stable/running-distributed.html for how to distribute the load over multiple CPU cores or machines
Stopped test
```

LOCUST

1. Về công cụ

Các chi tiết kỹ thuật thường gặp

Các class thường sử dụng

- User
- HttpUser
- TaskSet
- SequentialTaskSet

Hàm và phương thức

- wait_time
- on_start, on_stop
- Events và EventHook
- Configuration

LOCUST

2. Demo

Video demo 1: Hướng dẫn cài đặt

Môi trường lập trình (Python, IDE PyCharm hoặc VS Code)

Công cụ Locust

Link video: <https://youtu.be/guGoFRN-y9E>

LOCUST

2. Demo

Video demo 2: Demo tính năng

Nội dung Demo: Cài đặt và thực thi các tính năng nổi bật của Locust (Events, Custom Load Shape, Nested Taskset)

Kịch bản chính: Load test đến Google (www.google.com) và Doodle (www.google.com/doodle) cùng trang about của Doodle (www.google.com/doodles/about)

Kết quả: Chart kết quả load test đến 3 trang phía trên được phân bổ theo nested taskset cùng các load shape đã được custom.

Link Video: <https://youtu.be/btuWxuFEEdw>

LOCUST

2. Demo

Video demo 3: Demo đơn giản

Nội dung demo: Performance Testing chức năng mua hàng của Fahasa.com

Kịch bản chính:

- Truy cập trang sản phẩm (sách).

- Thêm sản phẩm đó vào giỏ hàng bằng API.

Kết quả testing:

- Kiểm tra giỏ hàng thấy số lượng hàng được thêm vào bằng đúng tổng số lần các User thêm vào giỏ hàng.

Video demo: <https://youtu.be/rgPsDfsQ-zc>

LOCUST

2. Demo

Video demo 4: Demo phức tạp

Nội dung demo: Gmail Performance Testing với Locust (tích hợp Selenium)

Kịch bản chính:

Đăng nhập Gmail

Mở email mới và soạn email

Gửi email

Kết quả: Các email được gửi thành công/soạn thành công nhưng chưa gửi/soạn chưa đúng format, cùng các biểu đồ kết quả (xem video)

Link video: https://youtu.be/lQYfxxuD_fE

THANK YOU



Nhóm C2 - Bulb Hacking Team

(PowerPoint Template by Alexander Aronowitz)

