Лабораторная работа 6. Разработка сценария нагрузочного тестирования с использованием Locust

Цель работы

Научиться разрабатывать сценарии нагрузочного тестирования с использованием **Locust**, тестируя API **OpenBMC** и открытое публичное API. Оценить производительность системы, проанализировать задержки и возможные точки отказа.

Теоретические сведения

- **Locust** – это инструмент для нагрузочного тестирования, который позволяет моделировать поведение множества пользователей, отправляющих HTTP-запросы к API или веб-приложениям.
  Основные возможности Locust:
  - Создание кастомных сценариев нагрузки на Python.
  - Запуск тестов с определенным числом виртуальных пользователей.
  - Мониторинг времени отклика API в реальном времени.
- **OpenBMC API** предоставляет Redfish-интерфейс для управления серверами, а также REST API для мониторинга и управления системой.
- **Открытое API** (например, JSONPlaceholder) будет использоваться для демонстрации работы Locust с внешними сервисами.

Ход работы

Часть 1: Настройка окружения

- Установите зависимости: locust
- Проверьте доступность OpenBMC (например, через curl):

```
curl -k -u root:0penBmc https://<BMC_IP>/redfish/v1/
```

- Выберите **публичное API** для тестирования, например:
  - JSONPlaceholder (https://jsonplaceholder.typicode.com/posts)
  - wttr.in (https://wttr.in/)

Часть 2: Разработка сценария нагрузочного тестирования:

- Написать Locust-скрипт (locustfile.py) с двумя классами нагрузки:
  - 1. Тестирование OpenBMC API
    - Запрос на получение информации о системе (/redfish/v1/Systems/system).
    - Запрос состояния питания (PowerState).
  - 2. Тестирование публичного API
    - Запрос списка постов на JSONPlaceholder (/posts).
    - Запрос погоды на wttr.in (https://wttr.in/Novosibirsk?format=j1).

Часть 3: Запуск нагрузочного тестирования

- Запустить Locust в режиме веб-интерфейса:

```
locust
```

- Перейти в браузере на http://localhost:8089.
- Ввести количество пользователей и скорость их запуска (например, 10 пользователей, 2 пользователя/сек).
- Запустить тестирование.

Часть 4: Анализ результатов

- Определить среднее время отклика API (метрики Locust).
- Проанализировать процент ошибок (если API не выдерживает нагрузку).
- Вывести графики нагрузки и оценить максимальную нагрузку, при которой API стабильно работает.

Отчет о лабораторной работе

В качестве результатов лабораторной работы необходимо приложить ссылка на GitHub репозиторий/ветку/директорию с реализованными тестами.