

ОБЗОР АВТОМАТИЗИРОВАННЫХ НАГРУЗОЧНЫХ ТЕСТОВ И ИНСТРУМЕНТОВ ДЛЯ ИХ РЕАЛИЗАЦИИ

Сорокина Татьяна Александровна

студент, Московский государственный технологический университет «СТАНКИН», РФ, г. Москва

Основная задача автоматизированного теста – автоматическое выполнение последовательности действий, описанных в сценарии тестирования. Грамотное построение структуры авто-теста и использование дополнительных инструментов позволяет автоматизировать не только их выполнение, но и запуск, инициализацию тестов, а также формирование отчета о прохождении тестов.

Основное внимание при обзоре авто-тестов будет уделяться структуре теста, определению функций, которые они автоматизируют, а также инструментам, которые позволяют задавать сценарии нагрузки.

В качестве первого примера рассмотрим авто-тест на языке «Python», задачей которого является проведение нагрузочного тестирования сервера «Json-server» с применением софта «Locust» [1]. Предварительным условием для реализации и запуска данного теста является настройка локального веб-сервера, а также подключение «Python». Структура авто-теста выглядит следующим образом. В начале определяются функции логина и логаута пользователей, а также задается «HTTP» объект сессии, с помощью которой задается нагрузка. Следующее несколько функций — запросы, за счет которых будет создаваться нагрузка. Далее — класс «UserBehavior», в нем будет описано поведение пользователя в тестируемом приложении. Свойству «tasks» передается словарь методов, которые будет вызывать пользователь и их частоту вызовов. В классе «WebsiteUser» задается модель поведения пользователя, а также минимальное и максимальное время ожидания между вызовами отдельных запросов каждым пользователем.

После запуска авто-теста в программе «Locust» отображаются результаты тестирования (рис.1): каждый из 10 созданных пользователей при старте зашел на главную страницу, список постов в среднем открывался в 2 раза чаще, чем писался комментарий, есть среднее и медианное время отклика для каждой операции, размер контента UI компонентов. Также можно увидеть графики нагрузки (рис.2):

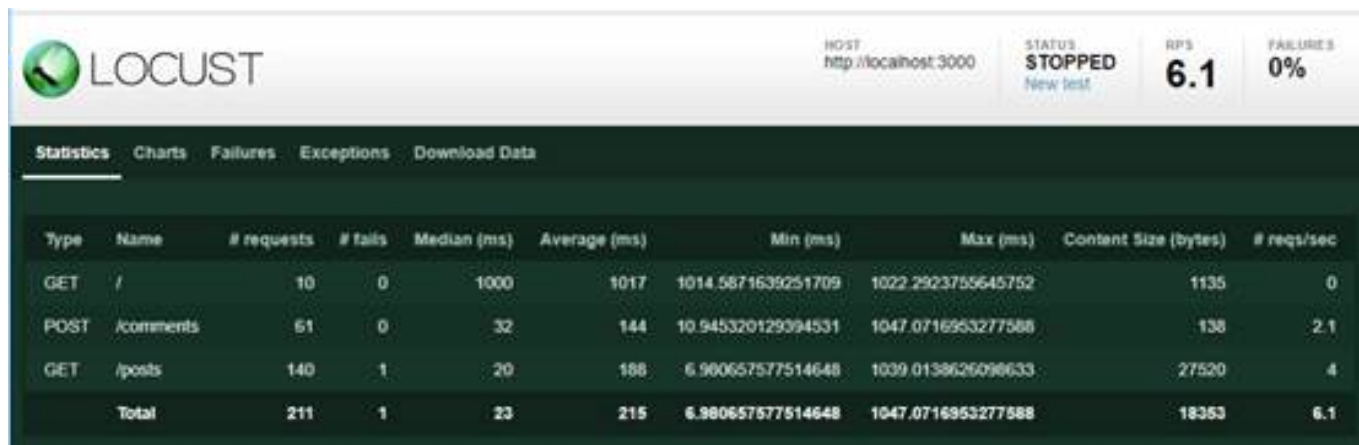


Рисунок 1. Отчет о результатах нагрузочного тестирования

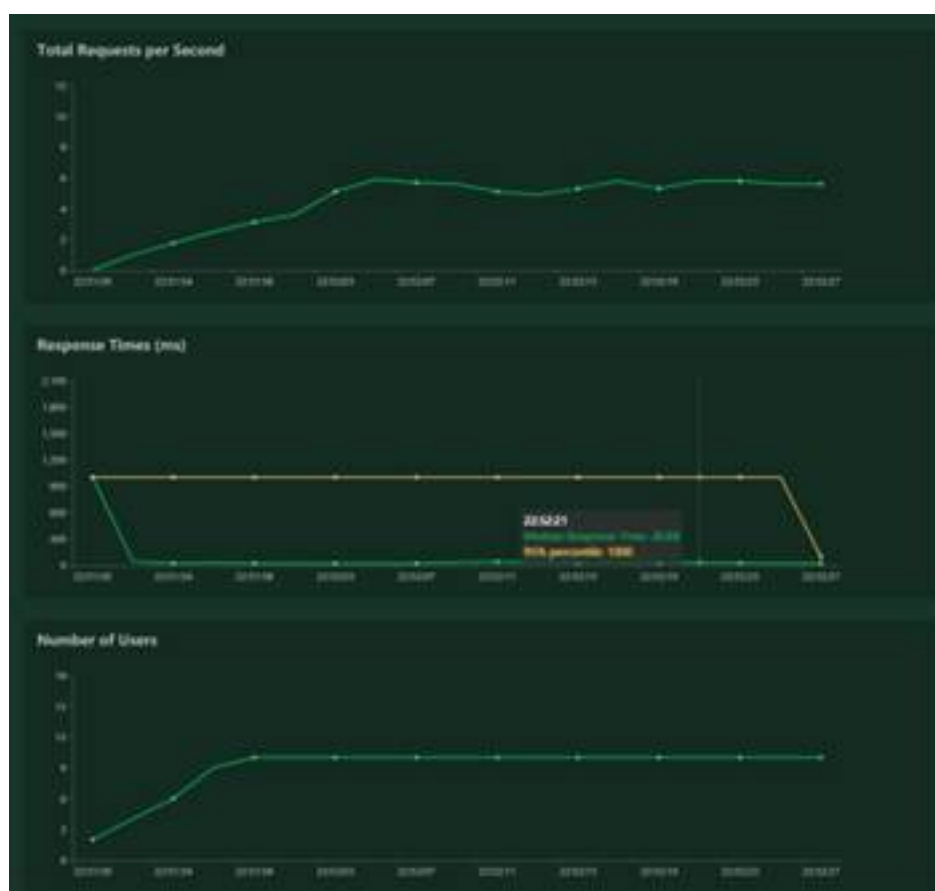


Рисунок 2. Графики нагрузки

В качестве второго примера рассмотрим авто-тест на языке «JavaScript», задачей которого является проведение нагрузочного тестирования браузера «Google*» с применением инструмента для тестирования «Postman» [2]. Предварительным условием для реализации и запуска данного теста является настройка клиента «Postman» и базовой конфигурации – установка количества итераций, порядка выполнения тестов, а также промежутка времени между запуском тестов.

Структура авто-теста выглядит следующим образом. В начале создается коллекция тестов под названием «Google* Apps – Load Testing». Затем, указываются условия, при которых должны выполняться определенные методы, в данном случае – выполнять запрос к страницам, если

предыдущий запрос имеет в качестве ответа статус код 200 (успех). Запуск данного теста осуществляется через командную строку. После завершения всех итераций теста, в консоли формируется статистика, где можно увидеть среднее время ответа на запросы (рис.3).

	executed	failed
iterations	2	0
requests	6	0
test-scripts	6	0
prerequisite-scripts	0	0
assertions	0	0
total run duration: 2.6s		
total data received: 3.52MB (approx)		
average response time: 385ms [min: 205ms, max: 815ms, s.d.: 209ms]		

Рисунок 3. Отчет о результатах нагрузочного тестирования

В качестве третьего примера рассмотрим авто-тест на языке C#, задачей которого является проведение нагрузочного тестирования локального веб-сервера с применением инструмента для тестирования «NBomber» [3]. Предварительным условием для реализации и запуска данного теста является настройка локального веб-сервера, а также подключение «NBomber». Авто-тест выполняет следующие задачи: конфигурирование числа пользователей, нагружающих «HTTP endpoint», генерация расширенной статистики по среднему времени обработки запроса и проценту ошибок.

Структура авто-теста выглядит следующим образом. В начале подключаются библиотеки, затем задается «HTTP» объект сессии, с помощью которой задается нагрузка. Запуск данного теста осуществляется через командную строку, в параметрах запуска указывается количество пользователей, задающих нагрузку. После завершения всех итераций теста, в консоли формируется статистика, где можно увидеть среднее время ответа на запросы и конфигурирование числа пользователей, нагружающих «HTTP endpoint» (рис.4).

scenario stats

scenario: 'Call DateNow Api'

duration: '00:01:00', ok count: 6000, fail count: 0, all data: 0 MB

load simulation: 'inject_per_sec', rate: 100, during: '00:01:00'

step	ok stats
name	call datenow
request count	all = 6000, ok = 6000, RPS = 100
latency	min = 0.18, mean = 9.51, max = 113.00, StdDev = 11.92
latency percentile	50% = 5.35, 75% = 10.65, 95% = 32.06, 99% = 64.29

status codes for scenario: Call DateNow Api

status code	count	message
200	6000	

Рисунок 4. Отчет о результатах нагрузочного тестирования

Для наглядного сравнения представленных ранее авто-тестов и инструментов, необходимых для их реализации, были составлены таблицы 1 и 2.

Таблица 1.

Сравнение существующих авто-тестов. Целевые функции

Авто-тест	Вид/виды нагрузочного тестирования	Целевая функция/функции	
Авто-тест, реализован-ный с помощью инструмента «locust»	Нагрузочное тестирование, тестирование производительности	1.Оценка поведения компонента системы в условиях увеличивающейся нагрузки 2. Оценка среднего и медиального времени отклика для каждой операции, размер контента UI компонентов	Р... данн... вып...
Авто-тест, реализованный с помощью инструмента «Postman»	Нагрузочные тестирование	Оценка поведения системы при увеличении нагрузки	Одн... вып...
Авто-тест, реализован-ный с помощью инструмента «NBomber»	Нагрузочные тестирование, тестирование производительности, тестирование надежности	1.Оценка поведения компонента системы в условиях увеличивающейся нагрузки 2. Оценка среднего времени отклика 3. Проверка способности системы выполнять свои функции в определенных условиях в течение заданного промежутка времени и при заданном количестве операций.	А... ге... вып...
Авто-тест, разработанный мной	Нагрузочное тестирование, стресс тестирование, тестирование стабильности, и производительности, объемное тестирование	1. Оценка максимальной производительности 2. Определение способности системы к регенерации 3. Выявление зависимости производительности от числа	А... ге... вып...

Авто-тест	Вид/виды нагрузочного тестирования	Целевая функция/функции	
		одновременно работающих пользователей 4. Проверять работоспособности системы при длительной нагрузке 5. Оценка влияния объемов данных на производительность	

Таблица 2.

Сравнение существующих авто-тестов. Нецелевые функции

Показатель	Авто-тест, реализован-ный с помощью инструмента «locust»	Авто-тест, реализованный с помощью инструмента «Postman»	Авто-тест, реализован-ный с помощью инструмента «NBomber»	Авто-тест, разработанный самостоятельно
Запуск теста	Осуществляется вручную	Осуществляется вручную	Осуществляется вручную	Осуществляется вручную
Выполнение теста	Осуществляется автоматически	Осуществляется автоматически	Осуществляется автоматически	Осуществляется автоматически
Поддержка системы контроля версиями	Не поддерживает	Не поддерживает	Поддерживает	Поддерживает
Графическое представление распределения нагрузки	Имеется	Не имеется	Не имеется	Имеется
Описание возникших ошибок	Не описываются, при неуспешном прохождении теста в консоли отображается ошибка	Не описываются, при неуспешном прохождении теста в консоли отображается ошибка	Не описываются, при неуспешном прохождении теста в консоли отображается ошибка	Ошибки описываются

Список литературы:

1. Сайт «Хабр» [Электронный ресурс]. Режим доступа: <https://habr.com/ru/companies/infopulse/articles/430502/>, свободный (дата обращения 01.04.2024 г.).

2. Сайт «Хабр» [Электронный ресурс]. Режим доступа: <https://habr.com/ru/company/otus/blog/677214/>, свободный (дата обращения 01.04.2024 г.).

3. Сайт «Хабр» [Электронный ресурс]. Режим доступа: <https://habr.com/ru/post/664824/>, свободный (дата обращения 01.04.2024 г.).

* По требованию Роскомнадзора информируем, что иностранное лицо, владеющее информационными ресурсами Google является нарушителем законодательства Российской Федерации – прим. ред.