



Нагрузочное тестирование

Практический курс по нагрузочному тестированию

Длительность курса: 88 академических часов

1 Введение в нагрузочное тестирование (НТ)

1 НТ: цели, события, показатели, основные этапы

Цели занятия:

объяснить, какие цели ставятся перед нагрузочным тестированием и какие события инициируют НТ;
объяснить, что такое показатели производительности системы и как их считать;
планировать нагрузочное тестирование и знать его основные этапы.

Краткое содержание:

цели и события НТ;
показатели производительности;
основные этапы НТ.

2 Профиль нагрузочного тестирования

Цели занятия:

проанализировать статистику использования системы и прогнозировать нагрузку;
составить профиль нагрузочного тестирования.
изучить способы составления профиля нагрузки и применения его к сценариям.

Краткое содержание:

статистика использования системы;
профиль НТ.

Домашние задания

1 Профиль нагрузочного тестирования

Цель: 1) Анализ статистики и составление профиля нагрузочного тестирования.

Требуется составить профиль НТ. Материалы - файл со статистикой (статистика_дз.xlsx) и файл автоанализа (Профиль_НТ_1.5_пустой.xlsx) прилагаются к материалам урока.

2) Расчёт параметров сценария нагрузочного тестирования

Требуется рассчитать параметры (VU и Pacing) сценария НТ.

Параметры:

* Список и интенсивности операций берутся из результатов задания "анализ статистики и составление профиля нагрузочного тестирования"

* Предполагается, что все операции выполняются не более 5 секунд (при одиночном прогоне без нагрузки)

Параметры сценария НТ:

* Разгон до 200%, после чего 5 ступенек по 20% каждая (220%, 240%, 260%, 280%, 300%).

Профиль НТ:

- Вставить статистику в файл анализа

- Отобрать дни, подходящие для составления профиля НТ

- Отобрать часы, для которых будет составляться профиль НТ

- Скорректировать список операций, попадающих в профиля

- Оценить итоговый профиль

Сценарий НТ:

Необходимо рассчитать параметры сценария НТ:

* Количество VU

* Pacing

Результаты предоставляются в XLSX-файле

3 Методика и отчёт по НТ

Цели занятия:

описать основные разделы методики нагрузочного тестирования;

определить типы отчётов и что можно в них указывать.

Краткое содержание:

методика НТ;

отчёт по НТ.

- 1 **Скрипты и сценарии НТ - 1: MF Performance center (PC) и Vugen часть 1/3**
- Цели занятия:**
- разработать скрипт web/http (web сайт).
- Краткое содержание:**
- настройка записи скрипта в LR Vugen и написание скрипта для тестирования web сайта;
выполнение параметризации данных в скрипте, как это влияет на его выполнение;
поиск динамических данных, получаемых от сервера, сбор их и использование в запросах;
настройка запуска скрипта.
-
- 2 **Скрипты и сценарии НТ - 1: MF Performance center (PC) и Vugen часть 2/3**
- Цели занятия:**
- создать сценарий нагрузки;
планировать и запускать тесты.
- Краткое содержание:**
- интерфейс MF PC;
создание сценария нагрузки;
настройка планировщика запуска тестов.
-
- 3 **Скрипты и сценарии НТ - 1: MF Performance center (PC) и Vugen часть 3/3**
- Цели занятия:**
- настроить LR Analysis;
добавить, объединить и настроить отображение графиков;
произвести расчёт производительности;
провести анализ ошибок.
- Краткое содержание:**
- интерфейс LR Analysis;
добавление графиков в сессию;
как и для чего объединять графики;
настройки отображения графиков;

производительность системы по результатам проведённого теста; ошибки в результатах выполнения, и причины их возникновения.

Домашние задания

1 LR, PC & ANALYSIS

Цель: В этом ДЗ вы разработаете скрипт нагрузочного тестирования, составите сценарий нагрузки, проведёте тестовое испытание, соберёте результаты тестов и подготовите краткий отчёт по ним. Выполнив это задание вы научитесь писать скрипты нагрузочного тестирования в LoadRunner VuGen, создавать сценарий тестирования в MF Performance Center и запускать нагрузочное испытание, анализировать результаты теста и составлять отчёт о результатах испытаний.

Домашнее задание состоит из трёх частей, их можно реализовывать по-очереди (пишите в slack для консультаций).

Принимается полностью готовое ДЗ.

====

ЧАСТЬ 1 (сделать один вариант):

1.1. Лёгкий вариант:

Разработать скрипт покупки билетов на <http://www.load-test.ru:1080/webtours/> от имени 2 разных пользователей (создать заранее), каждый раз выбираются случайные города (отправление, назначение). Можно использовать заранее наполненные параметры со случайным выбором элемента массива (настраивается в конкретном параметре).

===

Критерий успеха: при подстановке других аккаунтов и прогоне скрипта 10 раз у каждого из подставленных аккаунтов будет не меньше 1-го нового билета.

1.2. Обычный вариант:

Аналогично, вместе с тем, города выбираются из списка, возвращаемого сайтом. Для этого можно спарсить возвращаемый список городов, подставить случайный элемент получившегося

массива.

===

Критерий успеха: в скрипте реализован механизм корреляции города отправления и города назначения относительно случайного города, спаршенного из списка всех городов, предлагаемых сайтом.

При прогоне скрипта 10 раз как минимум у одного из подставленных аккаунтов будут билеты с различными городам отправления и назначения.

1.3. Трудный вариант:

Как в 1.2., при этом раз в несколько итераций все купленные пользователем билеты удаляются из internary.

===

Критерий успеха: билеты создаются. Если создать несколько билетов руками -- после 10 прогонов скрипт удалит все билеты, включая созданные руками.

ЧАСТЬ 2 :

2.1. Создать в структуре Perfomance Center папку со своей фамилией.

2.2. Внутри неё создать папку для скриптов и папку для сценариев.

2.3. Доработанный скрипт из части 1 сохранить в соответствующей папке.

2.4. Создать сценарий поиска максимальной производительности на следующих условиях:

а) ступени поиска тахperf составляют 10% от профиля;

б) время стабилизации ступени равно 10 минутам;

в) профиль (требования к) нагружаемой АС составляет (ОП\Ч): 9 000 входов, 13 500 покупок билетов, 450 отказов от всех приобретённых билетов;

г) SLA по времени ожидания (на 1 операцию) составляет 3 секунды;

д) замеряется время выполнения следующих операций: открытие страницы логина (i), вход в систему (ii), каждый этап покупки билета отдельно (от поиска до подтверждения, для каждого этапа -- своя тразнакция)(iii), *отмена всех билетов (iv).

*если реализовано в части 1

е) действия пользователей следуют в логическом порядке: для покупки билетов нужно войти в систему, для входа -- открыть страницу логина и

так далее.

ж) сценарий настроен: выбран уникальный хост в качестве генератора нагрузки, другой хост выбран в качестве контроллера, каждая итерация запускает скрипт 1 раз, нет логгирования и проксирования, скрипт продолжается в случае ошибки.

2.5. Запустить сценарий поиска максимальной производительности. Можно использовать timeslots.

ЧАСТЬ 3:

3.1. Изучите результаты вашего теста.

3.2. Загрузите "сырые" результаты для дальнейшего анализа на ПК.

3.3. Откройте результаты теста и добавьте графики Avg. Transaction Response Time (с наложенным графиком Running Vusers), Transaction per seconds (с наложенным графиком Running Vusers).

3.4. Оцените уровень найденной максимальной производительности и уровень пиковой производительности (если она была достигнута).

3.5. Подготовьте краткий отчет по шаблону из урока.

3.6. Сохраните рабочую сессию в Load Runner Analysis, сожмите в zip архив вместе с файлом отчёта, и отправьте ссылку на GoogleDrive (или аналог) в чат с преподавателем.

4 Скрипты и сценарии НТ - 2: Jmeter часть 1/3

Цели занятия:

записывать или создавать "вручную" скрипты нагрузочного тестирования в Jmeter, параметризовать и отлаживать их.

Краткое содержание:

Основные элементы Jmeter:

- ThreadGroup
- Sampler
- Extractors
- Listener
- Запись трафика
- Параметризация

Цели занятия:

вы научитесь работать с дополнительными элементами Jmeter:

- работа с транзакциями;
- ветвление логики скрипта;
- использование различных переменных: список / констант / генерируемых;
- модулирование скриптов (или переиспользование кода);
- отладка через Debugger;
- использование jsr223 для достижения целей, недоступных встроенными элементами jmeter.

Краткое содержание:

управление логикой скриптов:

- Controllers
- Test Fragments
- JSR223
- Конфигурационные элементы (Headers / Cookie / Variable)
- Функции Jmeter
- Отладка через Step-by-Step Debugger
- Запуск тестов в консольном и распределённом режимах

Домашние задания

1 Разработка скрипта и проведение теста в Jmeter

Цель: Домашнее задание делится на 2 части:

- Разработка скриптов
 - Проведение теста и оценка результата.
- ДЗ построено так, что бы вы могли выполнять их независимо.

Разработка скриптов:

- разработать скрипт нагрузочного тестирования в Jmeter

Проведение теста и оценка результатов:

- провести мини-тест
- оценить результаты

Домашнее задание считается успешно

выполненным в случае, если:

- Разработка скрипта: после выполнения скрипта вы видите в веб-интерфейсе сайта результаты работы скрипта.
- Проведение теста и оценка результата: Вы получили графики, отображающие интенсивности и времена отклика. Интенсивность совпадает с целевой.

```
=====
=====
=====Разработка
скрипта=====
===
=====
=====
```

Для выполнения скрипта необходимо зарегистрировать персонального пользователя на сайте

<http://webtours.load-test.ru:1080/webtours>

(резервный адрес)

<http://webtours.load-test.ru:1090/webtours>

Необходимо разработать скрипт, выполняющий следующие действия:

- Открытие страницы <http://webtours.load-test.ru:1080/webtours>
 - Вход в систему под вашим пользователем.
 - Переход на страницу "Flights", выбор случайного города отправления и прибытия.
 - В половине случаев скрипт должен покупать билеты "туда", а в половине случаев "туда-обратно"
 - Выбор случайного рейса по данному направлению из 4х предложенных (для билетов туда-обратно для 8ми)
 - Покупка билета
 - Переход на корневую страницу
- Элементы Jmeter, необходимые для выполнения задания:-----
- ThreadGroup
 - Cookie manager
 - Post Processor => Boundary extractor ИЛИ RegExp extractor
 - Использование \${переменных}
 - If controller (для покупки билета туда-обратно в половине случаев)

Задачи повышенной сложности:

- Обеспечить, что бы при покупке билета гарантированно не был куплен билет, где город отправления и прибытия совпадает.

- Разработать скрипт, переходящий на страничку с купленными билетами и отменяющий первый из них (важно учесть, скрипт должен работать вне зависимости от количества купленных билетов)

-----Элементы Jmeter, необходимые для выполнения задания:-----

- jsr223 Sampler (программирование на Groovy - считайте, что это java)

- Опционально: ForEach Controller

- while controller

=====

=====

=====Проведение

теста=====

=====

=====

- Необходимо создать 2 тред-группы, содержащей по одному http-семплеру:

Первая открывает ya.ru

Вторая открывает www.ru

- Необходимо подать следующую нагрузку:

В течение 5 минут разгоняемся до 100% профиля

В течение 10 минут подаём равномерную нагрузку в 100% профиля

В течение 5 минут разгоняемся до 120% профиля

В течение 10 минут подаём 120% профиля

100% профиля выглядит так:

ya.ru = 60 запросов в минуту

www.ru = 120 запросов в минуту

По итогам необходимо прислать JMX-файл + скриншоты графиков из Grafana, где видны И графики И запрос, который их формирует:

- Количество операций в минуту в течение всего теста

- Времена отклика в ходе теста

- цифрами написать, какие были времена отклика и сколько операций было выполнено на последней ступени теста (120%)



6 **Скрипты и сценарии НТ - 2: Jmeter часть 3/3**

Цели занятия:

создавать сценарий в котором нагрузка увеличивается ступеньками в нужном вам объёме. Кроме того, вы узнаете как разворачивать средства мониторинга работы ваших тестов.

Краткое содержание:

управление подаваемой нагрузкой и анализ результатов:

- Ultimate Thread Group;
 - Flow Control Action;
 - Constant Throughput Timer;
 - Throughput Shaping Timer;
 - Backend Listener;
 - Influx;
 - Grafana.
-

7 **Скрипты и сценарии НТ - 3: Gatling часть 1/2**

Цели занятия:

объяснить базовые действия: установка, базовые компоненты, вывод результатов в файл, типы сценариев, запись в influxdb и вывод графиков в Grafana.

Краткое содержание:

Components - Installation, Engine, Reporting, Recorder
Usage - Zip bundle, maven, sbt
Scala - Light intro, Gatling DSL
Basic Elements of Script
Scenario Definition
Checks
Pauses
Code reuse
Feeders
Assertions
Console run and generate report
Realtime monitoring

Цели занятия:

написание скриптов, собирать docker образ, запускать тесты в container'e; объяснить использование кастомных плагинов

Краткое содержание:

подключение сторонних плагинов;
сборка через sbt;
запуск тестов через Docker.

Домашние задания

1 Разработка скрипта и проведение теста в Gatling

Цель: Разрабатываем скрипт в средстве НТ, подготавливаем сценарий, проводим тест и анализируем результаты.

Для выполнения скрипта необходимо зарегистрировать персонального пользователя на сайте

<http://www.load-test.ru:1080/webtours>

(резервный адрес)

<http://79.165.34.170:1080/webtours>

Необходимо разработать скрипт, выполняющий следующие действия:

- Открытие страницы <http://www.load-test.ru:1080/webtours>
- Вход в систему под вашим пользователем.
- Переход на страницу "Flights", выбор случайного города отправления и прибытия.
- Выбор случайного рейса по данному направлению из 4х предложенных
- Покупка билета
- Переход на корневую страницу

Задание:

1. Реализовать ступенчатый тест от 0 - 100% (0 RPS и до момента деградации системы - увеличение числа ошибок или времени отклика операций) с шагом в 10% (от найденного максимума) используя Gatling (длительность теста любая, лучше не меньше 20 минут). Сборщик любой. В результате должен быть код сценария +

html отчет о тестах (или ссылка на grafana).
2. Реализовать тест надежности от 80% от максимальной производительности в течение часа.

9 **Автоматизация
НТ: CI / CD и
quality gate
часть 1/2**

Цели занятия:

определить критерии успешности тестов (Quality gate);
создать и запустить Job и Pipeline; использовать Jenkins-plugin для запуска сценариев MF Performance Center (PC);
параметризовать запуск Pipeline;
подключить Pipeline из Git; создать чек-лист запуска тестов через Pipeline;
добавить кастомные скрипты для сбора и анализа результатов теста в Job.

Краткое содержание:

критерии успешности тестов;
создание Jenkins Job;
настройка Jenkins-plugin для запуска сценариев MF PC;
добавление кастомных скриптов в Jenkins Job; создание Jenkins Pipeline;
создание Jenkins Pipeline на основе чек-листа запуска теста; подключение Pipeline из Git и параметризацию его исполнения.

10 **Автоматизация
НТ: CI / CD и
quality gate
часть 2/2**

Цели занятия:

создавать и использовать Gitlab CI пайплайны;
запускать Gatling тесты из Gitlab CI;
разбивать пайплайн на несколько стейджей с разными задачами;
выбрать для себя CI инструмент (Jenkins || Gitlab).

Краткое содержание:

попробуем создавать Gitlab CI пайплайны;
запустим Gatling из Gitlab CI;
сравним Jenkins и Gitlab.

- 11 **Разработка базовых эмуляторов внешних систем**
- Цели занятия:**
- разработать эмулятор REST web-сервиса, на wiremock.
развернуть эмулятор.
- Краткое содержание:**
- возможность замены SOAP сервиса, REST сервисом;
разработка эмулятора web-сервиса с использованием фреймворка wiremock.
запуск эмулятора.
-
- 12 **Скрипты и сценарии НТ - 4: k6.io часть 1/2**
- Цели занятия:**
- объяснить базовые действия: установка, компоненты, главная идея инструмента, вывод результатов, типы сценариев;
записать в influxdb и вывести данные в Grafana.
- Краткое содержание:**
- Installation;
ES6;
Virtual Users;
Test Structure;
Types of Tests;
Metrics;
Thresholds & Checks;
Running Test;
Results Visualization.
-
- 13 **Скрипты и сценарии НТ - 4: k6.io часть 2/2**
- Цели занятия:**
- настроить и запустить нагрузочные тесты через Github Actions вместе со сборкой микросервиса.
- Краткое содержание:**
- создание сервиса на node.js;
сборка сервиса в контейнер;
GitHub Actions - основные понятия;

написание workflow;
подключение и деплой на Heroku;
запуск деплоя и нагрузки из Github Actions;

Домашние задания

1 Разработка скрипта и проведение теста в k6.io

Цель: Разрабатываем скрипт в средстве НТ, подготавливаем сценарий, проводим тест и анализируем результаты

```
=====
=====
=====Разработка
скрипта=====
===
```

```
=====
=====
Для выполнения скрипта необходимо
зарегистрировать персонального пользователя на
сайте
http://www.load-test.ru:1080/webtours
(резервный адрес)
http://79.165.34.170:1080/webtours
```

Необходимо разработать скрипт, выполняющий следующие действия:

- Открытие страницы <http://www.load-test.ru:1080/webtours>
- Вход в систему под вашим пользователем.
- Переход на страницу "Flights", выбор случайного города отправления и прибытия.
- Выбор случайного рейса по данному направлению из 4х предложенных
- Покупка билета
- Переход на корневую страницу

```
=====
=====
=====Проведение
теста=====
=====
```

- Необходимо создать 2 сценария (реализовать все в 1 файле - скрипте), содержащей по одному http запросу:
Первая открывает ya.ru

Вторая открывает www.ru

- Необходимо подать следующую нагрузку:

В течение 5 минут разгоняемся до 100% профиля

В течение 10 минут подаём равномерную нагрузку
в 100% профиля

В течение 5 минут разгоняемся до 120% профиля

В течение 10 минут подаём 120% профиля

100% профиля выглядит так:

ya.ru = 60 запросов в минуту

www.ru = 120 запросов в минуту

Задание:

1. Разработать нагрузочные тесты на k6 (webtours)
и провести тестирование сайтов (ya.ru и www.ru) ->

Результаты в Influx, ссылку на Grafana (можно
приложить скриншоты с локального запуска) и
архив скриптов/ссылку на репозиторий

2. Разработать workflow на GitHub Actions и
запустить 1 из тестов на сайте из Actions ->
Результаты ссылкой на воркфлоу

1 **Мониторинг и узкие места: СУБД часть 1/2**

Цели занятия:

рассмотреть архитектуру реляционных СУБД и их узкие места.

Краткое содержание:

основные компоненты РСУБД;
нормализация (OLAP vs OLTP);
индексы;
транзакции, уровни изоляции;
план запроса.

2 **Мониторинг и узкие места: СУБД часть 2/2**

Цели занятия:

изучить инструменты нагрузочного тестирования и мониторинга РСУБД на примере Postgres.

Краткое содержание:

настройка мониторинга;
создание нагрузки на СУБД;
анализ результатов.

3 **Мониторинг серверной и сервисной инфраструктуры: основы, Часть 1**

Цели занятия:

получить представление о мониторинге в нагрузочном тестировании и научиться использовать стандартный инструментарий для его осуществления.

Краткое содержание:

разобраться, какие бывают классы метрик и как их отслеживать;
развернуть Prometheus и Grafana в docker, настроить их взаимодействие;
skonфигурировать prometheus и node exporter, дашборды в Grafana;
получить базовое представление о типах аппаратных метрик.

4 **Мониторинг серверной и сервисной инфраструктуры: Часть 2**

Цели занятия:

получить представление про роль логирования в мониторинге, на примере Elasticsearch;
узнать про базовые утилиты мониторинга Linux.

Краткое содержание:

логирование в NoSQL базу данных Elasticsearch;
исследуем её REST API;
пример синтетического наполнения БД;
утилиты мониторинга Linux (atop, sar, vmstat, iostat, ifstat).

5 Критерии качества НТ, регрессионное НТ и риски

Цели занятия:

определить слабые места текущего процесса по НТ и что можно улучшить;
определить критичные тесты для регрессионного НТ, в том числе при нехватке времени;
оценить риски производительности до начала НТ / по результатам НТ.

Краткое содержание:

регрессионное НТ;
риски и качество НТ.

Домашние задания

1 Подготовка заключения об НТ

Цель: Провести сравнение результатов 2-х версий системы, подготовить заключение о регрессионном тестировании

1. Проведите эталонный тест сайта, расположенного на одном из адресов webtours
2. Проведите регрессионный тест сайта на 2-м адресе webtours
4. Подготовьте заключение о релизе сайта, по шаблону, дайте ваши выводы, рекомендации.
*по желанию можете подготовить мини-отчёт или экспресс-отчет - ведь весь мониторинг в ваших руках!

Адреса webtours:

<http://webtours.load-test.ru:1080/webtours>

<http://webtours.load-test.ru:1090/webtours>

1 **Выбор темы и
организация
проектной
работы**

Цели занятия:

выбор темы и организация проектной работы.

Краткое содержание:

выбрать и обсудить тему проектной работы;
спланировать работу над проектом;
ознакомиться с регламентом работы над проектом.

Домашние задания

1 Проект

Цель: В проекте будет разработана методика и профиль нагрузочного тестирования, скрипты и сценарии, проведены тесты и подготовлен отчет о нагрузочном тестировании.

См. презентацию к уроку

2 **Консультация
по проектам и
домашним
заданиям**

Цели занятия:

получить ответы на вопросы по проекту, ДЗ и по курсу.

Краткое содержание:

вопросы по улучшению и оптимизации работы над проектом;
затруднения при выполнении ДЗ;
вопросы по программе.

**3 Защита
проектных
работ**

Цели занятия:

защитить проект и получить рекомендации экспертов.

Краткое содержание:

презентация проектов перед комиссией;
вопросы и комментарии по проектам.