

Автоматизация нагрузочного тестирования web-приложений с использованием инструмента Taurus

Кучер Илья Юрьевич

*Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых
студент*

Градусов Александр Борисович

*Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых
к.т.н., доцент, доцент кафедры вычислительной техники и систем
управления*

Аннотация

В данной статье рассматривается процесс автоматизации тестирования web-приложения с использованием инструмента Taurus. Любая информационная система должна обеспечивать стабильную работу под нагрузкой. Тестирование производительности системы позволяет оценить то как система ведет себя под определенной нагрузкой и, соответственно, отвечает ли программным требованиям, установленным заказчиками. В статье рассмотрены основные инструменты нагрузочного тестирования, а также описаны их ключевые достоинства и недостатки.

Ключевые слова: нагрузочное тестирование, производительность системы, автоматизация тестирования, инструменты тестирования производительности, сценарии тестирования.

Automation of load testing of web applications using the Taurus tool

Kucher Ilya Yurievich

*Vladimir State University named after Alexander and Nikolay Stoletovs
student*

Gradusov Alexander Borisovich

*Vladimir State University named after Alexander and Nikolay Stoletovs
Candidate of technical sciences, associate professor, associate professor of the
Department of computer engineering and control systems*

Abstract

This article discusses the process of automating testing a web application using the Taurus tool. Any information system should provide stable work under load. Testing the performance of the system allows you to assess how the system behaves under a certain load and whether it meets the software requirements set by

customers. The article discusses the main tools of load testing, and also describes their key advantages and disadvantages.

Keywords: load testing, system performance, test automation, performance testing tools, test scripts.

Любая коммерческая информационная система создается для того, чтобы приносить прибыль. А для того чтобы максимизировать прибыль, необходимо увеличить количество пользователей до наилучшего показателя и свести к минимуму потери. Если система нестабильна и неожиданно начинает «рушиться» при большом количестве посетителей, то это приводит лишь к оттоку возможных клиентов и к убыткам (реклама, подписки, продажа товаров или услуг). Именно поэтому нагрузочное тестирование web-приложений является не только полезным, но и финансово обоснованным.

Нагрузочное тестирование — определение или сбор показателей производительности и времени отклика программного продукта в ответ на внешний запрос с целью установления соответствия требованиям, предъявляемым к данной системе [1].

Одним из наиболее востребованных инструментов нагрузочного тестирования является приложение Apache JMeter. Apache JMeter — это информационная система с открытым исходным кодом. Приложение написано на языке Java и предназначено для загрузки информационных систем и измерения их производительности. Оно было первоначально разработано для тестирования веб-приложений, но с тех пор расширилось до других тестовых функций. Его можно использовать для симуляции большой нагрузки на сервер, группу серверов, сети или объекта, чтобы проверить его прочность или проанализировать общую производительность при разных типах нагрузки [2].

Как инструмент тестирования производительности JMeter имеет довольно много плюсов, но автоматизация процесса тестирования и интеграция с другими системами является довольно трудной задачей, также инструмент является довольно тяжелым в обучении. В этой статье будет рассмотрен инструмент Taurus — приложение автоматизации тестирования с открытым исходным кодом, которое расширяет и абстрагирует JMeter. Taurus упрощает процедуру создания, запуска и анализа тестов производительности.

Сам процесс нагрузочного тестирования не ограничивается процессами разработки и выполнения тестов нагрузки. Более того, они не охватывают даже 50% сопутствующих видов деятельности, которые включают [3]:

- определение и разработку сценариев нагрузочного тестирования;
- параметризацию и выполнение сценариев;
- анализ результатов сценария нагрузки;
- интеграция всех вышеперечисленных процессов в жизненный цикл разработки программного обеспечения;
- автоматизация перечисленных процессов.

Популярными альтернативами JMeter-а являются инструменты The Grinder или Gatling, каждый из которых имеет свои преимущества и недостатки. Все эти инструменты настраиваются по-разному, и формат вывода результатов нагрузочного теста также отличается.

Taurus в свою очередь является автоматизированной средой с открытым исходным кодом, которая в основном представляет собой уровень абстракции над JMeter (или Grinder, Gatling или Selenium, с дальнейшей поддержкой этих инструментов). Taurus обладает следующими преимуществами:

- простая настройка и обновление;
- легко читаемый, удобный для управления версиями и унифицированный DSL (предметно-ориентированный язык) для определения сценариев нагрузочного тестирования;
- возможность выполнять существующие тесты JMeter (или Grinder, Gatling или Selenium);
- возможность создавать новые тесты с нуля, используя DSL;
- возможность объединения нескольких существующих сценариев тестирования и / или тестов на основе DSL в один сценарий;
- отчетность в режиме реального времени;
- собственный формат результатов базового инструмента тестирования;
- интеграция со службой отчетов blazemeter;
- статистика консоли и псевдографические графики во время выполнения теста;
- совместимый с JUnit формат результатов теста
- позволяет определить гибкие критерии прохождения, провала и если результаты превышают пороговое значение, можно автоматически пометить тест(ы) как неудачные.

Процесс установки можно выполнить с помощью системы Python Package Management. Пользователям Windows, может потребоваться установить компилятор Microsoft Visual C ++ для Python 2.7.

После установки пакет Taurus создает папку конфигурации. Taurus создает файл 10-base.json в этой папке, и этот файл содержит исходную конфигурацию по умолчанию, которая может быть изменена в соответствии с вашими требованиями. Сохранение значений по умолчанию в папке конфигурации уменьшает размер файла конфигурации Taurus, так как настройки являются общими для всех тестов Taurus, которые выполняются с этого компьютера.

Taurus может использовать файлы конфигурации, написанные на языках JSON или YAML. Taurus добавляет отсутствующий DSL для JMeter и других поддерживаемых базовых инструментов. Например, способ определения сценария загрузки JMeter не очень очевиден, и необходимо выполнить некоторые вычисления в отношении настройки скорости прихода и замедления виртуальных пользователей, настроить количество итераций и

опции расписания, чтобы установить желаемую нагрузку, длительность теста и т. д. Это частично разрешается пользовательскими группами потоков, предоставляемыми через проект плагинов JMeter (Ultimate Thread Group или Stepping Thread Group), но другие ограничения сохраняются [4]. Более того, файлы JMeter .jmx на самом деле являются файлами XML, поэтому даже если вы сделаете небольшие изменения, система управления версиями не будет очень информативной, и будет довольно сложно выяснить природу новых изменений. Taurus упрощает вышеупомянутые процессы, предоставляя легко читаемые файлы конфигурации, которые могут быть интерпретированы даже кем-то без опыта работы с любым из поддерживаемых инструментов.

Работу инструмента рассмотрим на примере простого нагрузочного теста с 10 виртуальными пользователями, время разгона 1 минута (время за которые пользователи попадут на сайт), продолжительность 2,5 минуты. Нагружаемым сервисом является сайт example.com — этот домен создан для использования в качестве иллюстративных примеров, он может быть использован без предварительного согласования или запроса разрешения. Конфигурация посещения сайта example.com с запросами HTTP GET представлена на рисунке 1.

```
---
execution:
  concurrency: 10
  hold-for: 2m30s
  ramp-up: 1m
  scenario:
    requests:
      - url: http://example.com/
        method: GET
~
```

Рисунок 1 – Конфигурация сценария нагрузки

Вышеуказанная конфигурация YAML хранится в файле example.yml. В этом случае мы можем вызвать его командой `bzt example.yml`. После запуска механизм Taurus начнет выполнение теста, в том числе:

- загрузка последней версии JMeter (с последними плагинами);
- подготовка скрипта JMeter.jmx на основе предоставленного example.yml;
- старт с фактического теста JMeter;
- отображение статистики в реальном времени и основных графиков в текстовой консоли;
- печать резюме на консоль после завершения теста;
- сохранение результатов теста JMeter в формате, аналогичном тому, который предоставляют плагины JMeter.

Выполнение теста представлено на рисунке 2.

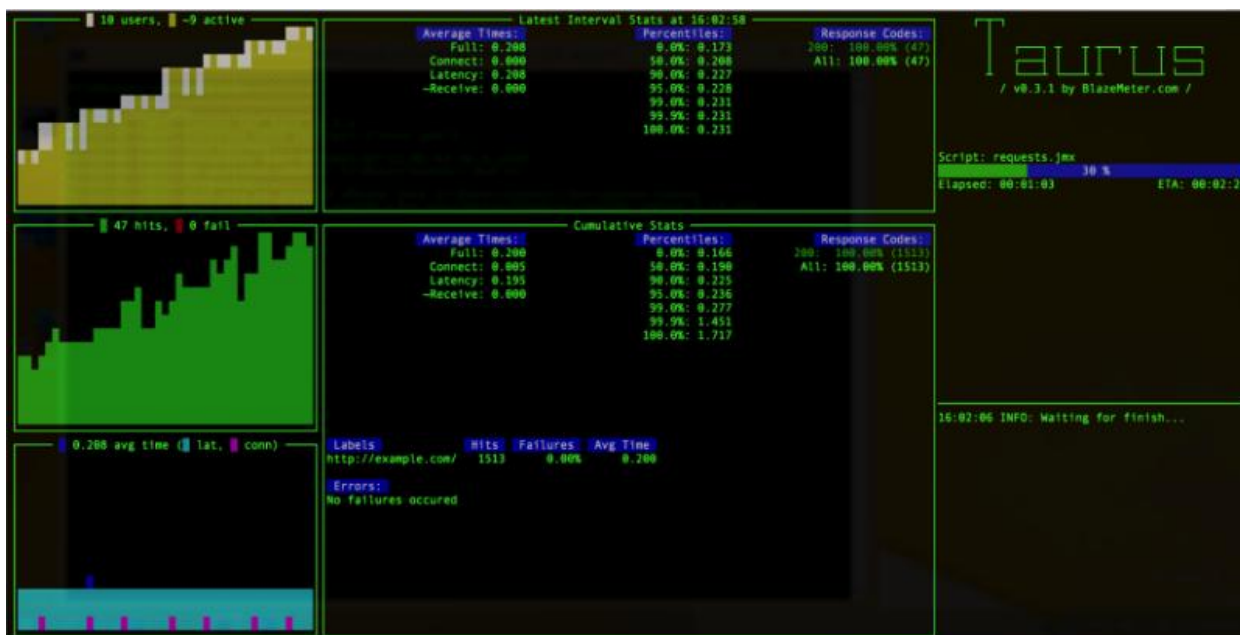


Рисунок 2 – Выполнения теста

Итоговый отчет после тестирования представлен на рисунке 3.

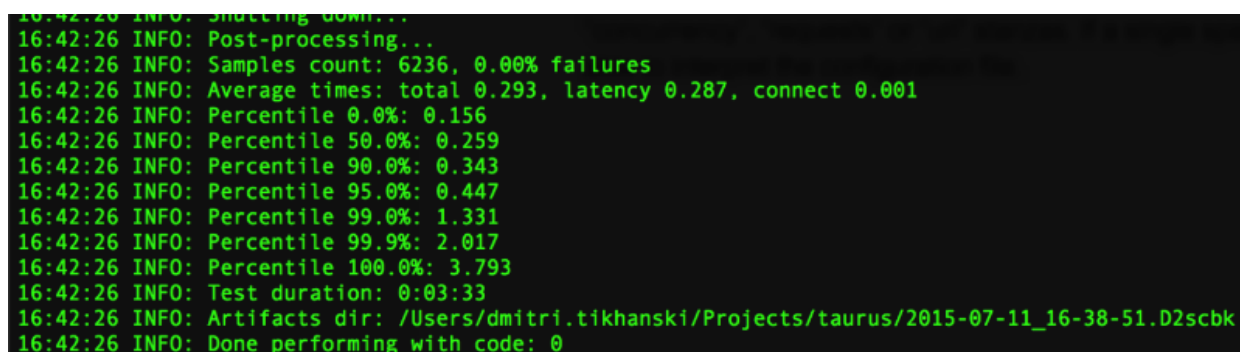


Рисунок 3 – Итоговый отчет

Первоначальные результаты теста можно проанализировать, просто посмотрев на вывод консоли. Samples count – количество прогонов теста, в нашем случае количество открытий сайта. Average times – среднее время отклика ресурса. Test duration – длительность теста.

Полная информация доступна в «Artifacts dir». Содержание «Artifacts dir» выглядит следующим образом:

- окончательный файл конфигурации Taurus в формате JSON, который содержит объединенные входные файлы, настройки по умолчанию, любые переопределенные свойства и т.д.;
- errors.jtl - файл результатов JMeter в формате XML со всеми включенными диагностическими полями, где можно увидеть полную информацию о запросах и ответах;
- example.yml - файл (ы) конфигурации YAML, предоставляемые через командную строку Taurus;

- `jmeter-bzt.properties` - любые свойства JMeter переопределяются;
- файлы логов журнала JMeter;
- основной файл результатов, куда входят ключевые показатели эффективности такие как время выполнения каждого запроса, продолжительность, ответные сообщения сервера, был ли пример успешным или нет, метрики подключения и задержки, номера активных потоков.

В итоге инструмент Tarus предоставляет унифицированный и упрощенный способ настройки, запуска автоматических тестов, позволяет представить результаты в наиболее эффективной форме.

Библиографический список

1. Нагрузочное тестирование программного обеспечения. URL: <http://unetway.com/tutorial/nagruzocnoe-testirovanie-programmnogo-obespecenia/> (дата обращения: 25.03.2019).
2. The Apache Software Foundation. Apache JMeter Documentation. URL: <http://jmeter.apache.org/usermanual/> (дата обращения: 24.03.2019).
3. ProTesting. Автоматизация тестирования ПО. URL: <http://www.protesting.ru/automation/performance.html> (дата обращения: 23.03.2019).
4. Jmeter-plugins. Ultimate Thread Group. URL: <https://jmeter-plugins.org/wiki/UltimateThreadGroup/> (дата обращения: 24.03.2019).