Васильев Леонид Федорович
магистрант
Ильин Денис Владимирович
магистрант
Петров Алексей Михайлович
магистрант
ФГБОУ ВО «Чувашский государственный
университет им. И.Н. Ульянова»
г. Чебоксары, Чувашская Республика

НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ КАК ЭТАП РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЙ

Аннотация: *в статье рассмотрено понятие «нагрузочное тестирование веб-приложений», описаны цели и этапы его проведения, разработан тест для определения быстродействия браузера.*

Ключевые слова: *нагрузочное тестирование, веб-приложение, тестовый стенд, скрипт, браузер.*

При создании веб-приложений [1–5] неотъемлемой частью разработки является нагрузочное тестирование, имитирующее работу большого количества пользователей на общем ресурсе. По его результатам предоставляется информация о способности системы правильно функционировать в случае превышения планируемых нагрузок при реальной эксплуатации [2].

Можно выделить несколько основных целей нагрузочного тестирования:

1) оценка производительности приложения на этапе разработки;

2) оценка производительности приложения на этапе выпуска;

3) оптимизация производительности приложения, включая настройки серверов и оптимизацию кода;

4) подбор соответствующей для данного приложения аппаратной (программной платформы) и конфигурации сервера.

Для проведения нагрузочного тестирования веб-приложений существует множество как платных, так и бесплатных систем. Наиболее популярными являются WAPT, Tsung, Apache JMeter, Joe Dog Siege и Apache HTTP server benchmarking tool. Недостатками бесплатных систем являются: ограниченное количество пользователей; невозможность выполнять сложные скрипты; высокая требовательность к ресурсам сервера.

Для проведения нагрузочных тестирований можно также разработать собственную программу. В любом случае проведение нагрузочного тестирования состоит из нескольких этапов: анализ требований; анализ целевой аудитории; определение базового профиля нагрузки; разработка моделей нагрузки.

При анализе требований необходимо определить основные критерии успешности проведенных тестов, а именно выделить следующие характеристики:

– время отклика – время необходимое для открытия страницы или получения ожидаемого результата;

– интенсивность – число запросов в секунду (Qps);

– используемые ресурсы – загрузка процессора, количество используемой памяти, дисковый и сетевой ввод-вывод данных;

– максимальное количество пользователей – число пользователей, способных работать с системой в условиях заданной конфигурации;

– некоторые другие метрики, связанные с работой бизнес сценариев, например, количество бизнес операций в единицу времени, время выполнения бизнес операции и т. д.

Заданные в требованиях характеристики, будут являться *базовыми нагрузочными точками* тестируемого приложения. Все получаемые результаты будут сравниваться с ними для принятия решения о завершении тестирования либо дальнейшем профилировании производительности.

Общение непосредственно с пользователями, наблюдение за их действиями, помогает более точно понять, как будет происходить работа с программным

обеспечением, какие части приложения наиболее критичны и требуют особого внимания. В результате получатся:

– четко выделенные группы пользователей (например, администраторы, операторы, зарегистрированные клиенты, новые пользователи и т. д.);

– описание сценариев тестирования, на основе реального поведения пользователей из каждой группы;

– график распределение выполнения бизнес операций в течении дня.

Выделив на предыдущем шаге группы пользователей, бизнес сценарии, требования к системе, можно приступить к созданию профиля нагрузки – набору операций и интенсивности их выполнения для каждой пользовательской группы. Зная количество пользователей в каждой группе, их нужно распределить для выполнения требуемых операций, сохранив необходимую интенсивность.

В зависимости от вида и целей проводимого тестирования, следует разрабатывать разные модели нагрузки. Разным может быть: количество пользователей и интенсивность запросов; старт самих пользователей: последовательно по одному, ступенчато группами или же запуск всех сразу; длительность сценария от 10 минут до нескольких часов или даже дней.

Проще всего проводить изменения, варьируя параметрами в базовом профиле нагрузки и инструменте для тестирования производительности (например, в HP LoadRunner Controller).

Четкое следование всем вышеописанным инструкциям по разработке моделей нагрузки, позволит:

– провести дополнительный анализ и тестирование требований по производительности;

– уточнить параметры и характеристики производительности;

– получить более четкое представление о работе системы в целом;

– получить на выходе план предстоящих работ, связанных с нагрузочным тестированием;

– определить предельный объем данных системы (с сохранением приемлемой производительности);

– определить предельное количество пользователей (групп) системы (с сохранением приемлемой производительности);

– определить ресурсоёмкие операции или сценарии (для дальнейшего профилирования системы);

– отслеживать эффект от вводимых оптимизаций системы при регулярных измерениях производительности, используя разработанные и проверенные модели нагрузки.

В итоге, имея готовые разработанные модели нагрузки, можно переходить к следующему этапу подготовки к нагрузочному тестированию – конфигурации тестового стенда. Тестовый стенд можно разделить на несколько частей: виртуальные пользователи; метрика; скрипты; отчеты.

Для каждого проекта составляются тесты по определенному алгоритму. Скрипты пишутся на разных языках, но основными являются C# и java.

Процесс проведения нагрузочного тестирования ведет к выполнению полного цикла работ над разработкой модели нагрузки, сбора тестового стенда, написания сценариев тестов, проведения и анализа тестов. При сборе тестового стенда требуется выбрать браузер. Для определения самого эффективного браузера было решено проверить каждый из них на небольшом тесте:

```
for (int i=0; i<10; i++)
{ using (var driver = new ChromeDriver())
{ driver.Navigate().GoToUrl(«http://testing-ground.scraping.pro/login»);
// Get the page elements
var userNameField = driver.FindElementById(«usr»);
var userPasswordField = driver.FindElementById(«pwd»);
var loginButton = driver.FindElementByXPath("//input[@value='Login']");
// Type user name and password
userNameField.SendKeys(«admin»);
userPasswordField.SendKeys(«12345»);
// and click the login button
loginButton.Click();
```

```
// Extract the text and save it into result.txt
var result = driver.FindElementByXPath("//div[@id='case_login']/h3»).Text;
File.WriteAllText(«result.txt», result);
// Take a screenshot and save it into screen.png
driver.GetScreenshot().SaveAsFile(@»screen»+i+”.png», ImageFormat.Png);
driver.quit()
}}
```

По данным теста (таблица 1) можно судить о быстродействии браузера.

Таблица 1

Скорость выполнения теста различными браузерами

Имя браузера	Количество испытаний	Среднее время, с
HtmlUnit	10	1,256
Firefox 4	10	1,543
Internet Explorer 9	10	1,845
Chrome	10	1,374
Opera 10	10	1,586

По результатам тестирования для сборки стенда было выбран популярный браузер Chrome со средним временем выполнения теста 1,374 с.

Описанный подход проведения нагрузочного тестирования является универсальным и может быть использован при разработке веб-приложений различного уровня сложности, в том числе автоматизированной системы проведения турниров по спортивному программированию.

Список литературы

1. Албутов К.В. Система электронной подачи абитуриентов / К.В. Албутов, О.А. Лобастова // Информатика и вычислительная техника: Сб. науч. трудов. – Чебоксары: Изд-во Чуваш. ун-та, 2016. – С. 10–12.

2. Васильева А.С. Система тестирования программного обеспечения интернет-магазина «Викимарт» / А.С. Васильева, Л.А. Павлов // Информатика и вычислительная техника: Сб. науч. трудов. – Чебоксары: Изд-во Чуваш. ун-та, 2016. – С. 55–58.

3. Краснов В.А. Реализация удаленного доступа к системе «Умный дом» / В.А. Краснов, А.А. Андреева // Информатика и вычислительная техника: Сб. науч. трудов. – Чебоксары: Изд-во Чуваш. ун-та, 2016. – С. 108–113.

4. Петров А.М. Разработка веб-приложения «Электронный журнал преподавателя» / А.М. Петров, О.А. Лобастова // Информатика и вычислительная техника: Сб. науч. трудов. – Чебоксары: Изд-во Чуваш. ун-та, 2016. – С. 144–149.

5. Шоркина А.Л. Разработка веб-приложения для подготовки к ЕГЭ по информатике / А.Л. Шоркина, О.А. Лобастова // Информатика и вычислительная техника: Сб. науч. трудов. – Чебоксары: Изд-во Чуваш. ун-та, 2016. – С. 123–128.

6. Булат А. Модель нагрузки как отправная точка при тестировании производительности [Электронный ресурс]. – Режим доступа: http://alexeybulat.blogspot.com/2010/09/blog-post.html (дата обращения: 28.03.2017).