

Разработка плана нагрузочного тестирования для web-приложения

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 статье рассматривается процесс разработки плана нагрузочного тестирования web-приложения. Нагрузочное тестирование позволяет определить, как и с какой скоростью работает программа при определенной нагрузке. Одним из основных этапов нагрузочного тестирования является разработка плана тестирования. В статье проводится анализ особенностей плана нагрузочного тестирования, на примерах рассматриваются ключевые этапы разработки плана.

Ключевые слова: нагрузочное тестирование, производительность системы, план тестирования, стабильность программы, виртуальный пользователь, ожидаемая нагрузка.

Develop a stress test plan for a web-application

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

The article discusses the process of developing a load testing plan for a web application. Load testing allows you to determine how and at what speed the program works under a certain load. One of the main stages of load testing is the development of a test plan. The article analyzes the features of the load testing plan, the key stages of the plan development are considered on the examples.

Keywords: load testing, system performance, test plan, program stability, virtual user, expected load.

В связи с постоянным усложнением и развитием web-приложений, а также возрастающей нагрузкой на них, одной из важнейших проблем разработчиков становится обеспечение высокой производительности своего продукта. В исследованиях посвященных данной теме говорится о том, что от производительности и стабильности сайта непосредственно зависит количество посетителей, рост продаж и увеличение трафика. Любая программа должна работать под нагрузкой как можно более длительное время. Разного рода сбои и отказы системы все чаще приводят к потере клиентов, убыткам и другим неприятным последствиям.

Поэтому одним из важнейших этапов разработки программного обеспечения является нагрузочное тестирование системы, которое дает возможность понять, каким образом программа ведет себя в различных стрессовых ситуациях, с какой скоростью работает под заданной нагрузкой.

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].

Основной целью тестирования нагрузки является мониторинг системы и снятие показателей производительности, путем создания в системе определённой ожидаемой нагрузки, с использованием виртуальных пользователей. Зачастую в процессе тестирования используют идентичное с реальной системой, программное и аппаратное обеспечение. Критериями успешности данного вида тестирования обычно являются требования к производительности программного продукта, которые должны быть сформированы и задокументированы до начала разработки основных модулей системы, т.е. на стадии разработки требований к информационной системе.

До начала тестирования производительности проводятся подготовительные работы, а также функциональное тестирование, тестирование usability, верстки и безопасности. Первым этапом тестирования нагрузки является разработка тест плана.

План тестирования — это документ, который описывает весь предстоящий объем работ по тестированию системы [2]. В документе необходимо описать тестируемый объект, привести список функций и компонентов системы, описать стратегии тестирования и график работ, критерии начала и окончания тестов, оборудование и программные средства, которые будут использоваться в процессе работы, а также специальные знания необходимые для работы. Документ также должен содержать в себе оценку всех возможных рисков и предполагаемых вариантов их разрешения, детальное описание окружения тестируемой системы.

В методологии RUP (Rational Unified Process) приводится список пунктов, которые должен включать в себя план тестирования нагрузки. В

данном списке содержаться следующие пункты: история изменений документа, используемые термины, цели тестирования, архитектура тестируемой системы, модель нагрузки, описание стратегии тестирования, критериев успешности теста, требования к тестовому стенду, ресурсы, используемые в работе и документы, которые подлежат сдаче по итогам тестирования [3].

Все изменения документа необходимо фиксировать в таблице изменений. Отдельным пунктом плана тестирования является список терминов и определений, которые будут использоваться во всем документе. Это такие термины как: производительность, виртуальный пользователь, нагрузочная точка, интенсивность выполнения операции, нагрузка, итерация и т.д.

На первом этапе разработки плана необходимо выделить и определить основные цели тестирования. Сформулировать цели можно следующим образом:

- оценить нагрузку на систему, т.е. определить пиковую, и среднюю нагрузки, которую может выдержать система;
- оценить стабильность работы системы при бесперебойной работе;
- выяснить достаточно ли серверу ресурсов (оперативной памяти, ресурсов центрального процессора и т.д.), чтобы обрабатывать ожидаемый трафик;
- выяснить есть ли особо ресурсоемкие страницы или вызовы API;
- определить с достаточной ли скоростью сервер реагирует на запросы пользователя;
- протестировать устойчивость системы при длительных нагрузках.

На следующем этапе описывается архитектура системы, схемы промышленного и тестовых стендов. При перечислении программно-аппаратных средств, используемых в процессе тестирования, детально описываются конфигурации, настройки и состояния промышленного и тестовых стендов (процессор, объем оперативной и постоянной памяти, операционная система, которая установлена на сервере и т.д.). Для достижения наиболее точных результатов, необходимо, чтобы тестовый стенд имел как можно более приближенную конфигурацию и аппаратные средства к той, которая используется реальными клиентами. На этом этапе приводится краткое описание тестируемой системы (описываются страницы сайта, глобальные и сквозные элементы, общие свойства страниц и их особые состояния, указываются разделы, крупные блоки, типы элементов).

Далее для тестируемой системы разрабатываются и описываются профили нагрузки. Профиль нагрузки — набор моделируемых операций и совместно с данными об их производительности. Т.е. для создания данных профилей необходимо сформировать список операций, подлежащих тестированию, ролей пользователей, собрать данные о требуемой производительности по каждой операции, которая войдет в разрабатываемый профиль, а также проанализировать и описать все возможные взаимосвязи

между выполняемыми операциями. Примером может служить нагрузочная модель, состоящая из N профилей нагрузки, она составляется на основании собранной или предоставленной заказчиком статистики. В таком случае каждая из операций должна будет моделироваться отдельным нагрузочным скриптом. Соответственно, каждый из этих скриптов будет выполняться отдельной группой виртуальных пользователей. Группы виртуальных пользователей должны работать отдельно и независимо друг от друга. Исходя из этого, все операции на протяжении тестирования будут выполняться параллельно и оторвано друг от друга. В данной ситуации для каждой из групп пользователей необходимо привлекать N ролей.

После определения нагрузочных профилей тестировщику необходимо произвести расчет базовой нагрузочной точки. Нагрузочная точка — рассчитанное (либо назначенное клиентом) количество виртуальных пользователей в группах, которые выполняют операции с определенными интенсивностями [4]. Расчет нагрузочной точки происходит путем выделения групп пользователей и определения интенсивностей, с которыми пользователи в данных группах будут выполнять необходимые операции. Количество виртуальных пользователей в каждой группе должны задаваться на основании информации полученной от заказчика.

На следующем этапе должна быть описана общая стратегия, на основании которой будет производиться тестирование, а также нагрузочные точки и сценарии, которые будут использоваться в процессе тестирования производительности. На протяжении работы могут быть добавлены дополнительные точки, которые рассчитаются исходя из базовой. Также на данном этапе выделяются ключевые критерии успешности прохождения каждого из тестов и всего процесса в целом. Затем проводится мониторинг аппаратных серверов, определяются метрики и способы, с помощью которых будет производиться мониторинг серверов и ресурсов. Необходимо выделить и классифицировать те случаи, при которых результаты тестов не будут приниматься к рассмотрению.

Дополнительно в тест плане могут указываться требования к тестируемому окружению, перечисляться роли, которые задействовались в процессе тестирования и уровни ответственности на разных этапах, описываться документы, которые использовались при разработке плана тестирования.

Разработка плана тестирования производительности является одним из важнейших этапов разработки программного обеспечения, его правильное создание, а также своевременное обновление, ускоряет и систематизирует весь процесс тестирования. План нагрузочного тестирования позволяет согласовывать объёмы и стратегию тестирования со всеми участниками команды проекта, производить своевременное планирование и учет ресурсов, необходимых в процессе тестирования, правильно расставлять приоритеты задач и заблаговременно учесть возможные риски.

Библиографический список

1. Нагрузочное тестирование. Тестирование работы ресурса при высоких нагрузках. URL: <https://www.pentestit.ru/audit/loadtesting.html> (дата обращения: 11.04.2018).
2. Рекомендации по написанию тест плана. URL: <http://www.protesting.ru/testing/plan.html> (дата обращения: 12.04.2018).
3. План нагрузочного тестирования. URL: <http://pandia.ru/text/77/278/61792.php> (дата обращения: 12.04.2018).
4. Куликов С.С. Тестирование программного обеспечения. Базовый курс. М.: Четыре четверти, 2015. С. 20.