

ВИДЫ ТЕСТИРОВАНИЯ ПРОИЗВОДИТЕЛЬНОСТИ WEB-ПРИЛОЖЕНИЙ

Бобровников В.И., Терешко И.С., Попова Ю.Б.
Белорусский национальный технический университет, г. Минск

Ежедневно многомиллионная пользовательская аудитория сталкивается с проблемами медленной работы web-приложений. Поэтому перед выходом на рынок программный продукт должен быть протестирован на скорость, стабильность, масштабируемость при различных условиях нагрузки. При игнорировании данных тестов существует большая вероятность встречи с такими трудностями, как медленная работа системы при подключении нескольких пользователей, и, как следствие, снижение уровня продаж данного продукта и ухудшение репутации производителя [1].

Тестирование производительности (англ., Performance testing) включает в себя ряд многочисленных тестов, которые позволяют проанализировать различные аспекты системы. Типы тестирования производительности представлены на рисунке 1 и описаны ниже.



Рисунок 1 – Типы тестирования производительности [2]

1. Нагрузочное тестирование (англ., Load testing). Данный тип тестирования имеет такие альтернативные названия, как Endurance testing (испытание на выносливость) и Volume testing (тестирование объема), и предназначен для проверки работы системы с постоянным увеличением нагрузки до тех пор, пока временная нагрузка не достигнет своего порогового значения. Здесь увеличение нагрузки подразумевает, например, увеличение количества одновременных пользователей, транзакций. Основная цель нагрузочного тестирования – отслеживание времени отклика

и остаточной мощности приложения при положительной работе системы под большой нагрузкой. При проведении нагрузочного тестирования, как правило, используется следующий набор показателей:

1) Емкость системы, т.е. предельная нагрузка, при которой система работает корректно.

2) Время отклика, т.е. скорость, с которой пользователи системы получают отклики на свои запросы.

3) Отказоустойчивость. Этот показатель заключается в том, насколько можно положиться на этот сервис, как он быстро восстановится в случае сбоя, как долго сможет работать до отказа.

4) Доступность – это время, в течение которого сервер находится в безотказном состоянии. Этот показатель вытекает из времени восстановления и времени наработки на отказ.

Существует довольно большой перечень инструментов для проведения нагрузочного тестирования, самым распространенным из которых является Apache Jmeter, способный проводить нагрузочные тесты для JDBC-соединений, FTP, LDAP, SOAP, JMS, POP3, IMAP, HTTP и TCP. Он является бесплатным инструментом и независимой платформой, имеет понятную структуру тестов, поддерживает работу с переменными, регулярными выражениями, парсинг JSON, работу с cookie, визуализацию результатов теста. Главным недостатком инструмента является отсутствие полноценного руководства пользователю.

2. Стрессовое тестирование (англ., Stress testing). Данный термин, известный также как Fatigue testing (испытание на усталость), служит для проверки стабильности работы программного обеспечения (ПО), когда аппаратные ресурсы недостаточны (например, процессор, память, дисковое пространство и так далее). Суть стрессового тестирования состоит в нагрузке на систему большим количеством пользователей/процессов, которые не могут быть обработаны системными аппаратными ресурсами. Оно должно обеспечить стабильность приложения посредством тестирования за пределами его пропускной способности. Основная идея – определение отказа системы и отслеживание ее восстановления [3].

3. Тестирование шипами (англ., Spike testing). Данный тип тестирования является подвидом стрессового тестирования. Его проведение организовывается для проверки эксплуатационных характеристик, когда тестируемая система в течение коротких периодов времени подвергается нагрузке, которая во много раз превышает ожидаемую.

4. Тестирование стабильности (англ., Endurance testing). В некоторых случаях тестирование стабильности также называют Soak testing (тестирование на выдержку). Тестирование стабильности подразумевает тестирование системы с ожидаемым объемом нагрузки в течение длительного периода времени. В качестве примера рассмотрим случай, когда система, рассчитанная на работу в течение 3 часов, выдерживает

работу в течение 6 часов. Необходимость данной проверки заключается в определении остаточной мощности системы. Чаще всего выполняются тесты для проверки такого поведения системы, как утечка памяти, сбой системы или случайное поведение [4].

5. Тестирование масштабируемости (англ., Scalability testing). Данный тип тестирования предназначен для проверки способности ПО по увеличению и уменьшению масштаба любых его нефункциональных возможностей. При этом приложение должно выдерживать пользовательскую нагрузку, поддерживать необходимое количество транзакций и объем данных [5].

6. Тестирование объема (англ., Volume testing). Задачей данного типа тестирования является получение оценки производительности при увеличении объема данных в базе данных приложения. При этом происходит измерение времени выполнения выбранных операций при определенных интенсивностях выполнения этих операций; может производиться определение количества пользователей, одновременно работающих с приложением.

Таким образом, для корректной работы системы тестирование производительности крайне необходимо. Оно играет ключевую роль в тестировании жизненного цикла программного обеспечения. Это вызывает повышение удовлетворения пользователей/клиентов и снижение вероятности некорректной работы/отказа продукта. Что доказывает глобальную роль проведения тестирования производительности перед выпуском продукта на рынок.

Литература

1. Савин, Р. Тестирование dot com или Пособие по жестокому обращению с багами в интернет-стартапах / Р. Савин. – Москва: Издательство «ДЕЛО», 2007. – 311 с.

2. Куликов, С. Тестирование программного обеспечения. Базовый курс / С. Куликов. – ЕРАМ Systems, 2018. – 296 с.

3. Стресс-тестирование [Электронный ресурс]. – Режим доступа <http://devopswiki.net/index.php> – Дата доступа: 10.03.2018.

4. Тестирование стабильности [Электронный ресурс]. – Режим доступа <http://devopswiki.net/index.php> – Дата доступа: 10.03.2018.

5. Тестирование масштабируемости [Электронный ресурс]. – Режим доступа <http://devopswiki.net/index.php> – Дата доступа: 10.03.2018.