



Тестирование производительности



Тестирование производительности

Тестирование производительности – тестирование, целью которого служит определение скорости работы вычислительной системы или ее частей под определёнными нагрузками.

Служит проверкой или подтверждением атрибутов качества системы, например надёжность системы или её масштабируемость.

Масштабируемость – способность системы справляться с увеличением рабочей нагрузки при добавлении ресурсов.

Тестирование производительности

Данный вид тестирования позволяет находить возможные уязвимости и недостатки в системе с целью предотвратить их отрицательное влияние на работу программы.

Необходимые параметры системы в определенной среде можно тестировать с помощью:

1. Определения рабочего количества пользователей приложения
2. Измерение времени выполнения различных операций системы
3. Определение производительности приложения при различных степенях нагрузки
4. Определение допустимых границ производительности программы при разных уровнях нагрузки



Тестирование производительности

Тестирование производительности проводится для обеспечения бесперебойной работы и снижения стоимости владения ПО на всех этапах жизненного цикла решения.

Существует 9 этапов тестирования производительности.



9 Этапов тестирования производительности

1 этап: Анализ системы и рассмотрение требований.

Главное условие данного этапа – полностью законченное приложение или система. Если процесс разработки не завершен, то полученные данные отразят недействительные результаты.

Следующее условие для данного этапа – правильно настроенная конфигурация ПО.



9 Этапов тестирования производительности

1 этап: Анализ системы и рассмотрение требований.

Анализ системы включает в себя изучение свойств системы, её особенностей и режима работы. С помощью анализа системы можно:

1. Воспроизводить максимально объективные шаблоны поведения пользователей и профиля нагрузки
2. Определить необходимое количество тестовых данных
3. Настраивать способы мониторинга системы
4. Выявить потенциально слабые места ПО



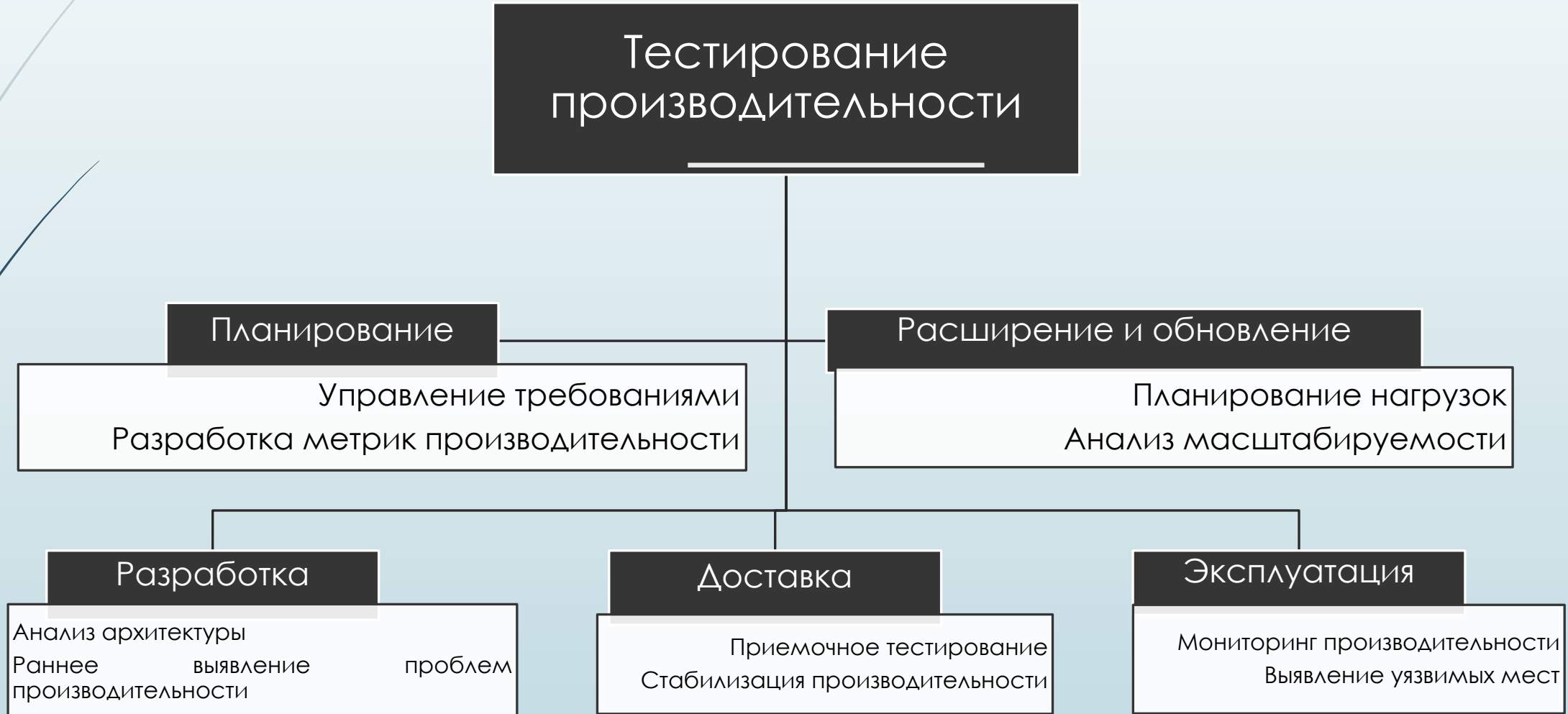
9 Этапов тестирования производительности

1 этап: Анализ системы и рассмотрение требований.

Требование – это критерий, которому система должна соответствовать с технической точки зрения.

Подобранные на начальных этапах требования необходимо сравнивать с полученными результатами чтобы оценить поведение ПО целиком и по модулям, для выявления и устранения слабых мест системы.

9 Этапов тестирования производительности





9 Этапов тестирования производительности

1 этап: Анализ системы и рассмотрение требований.

Критерии для измеряемых метрик:

1. Количество пользователей – программа, которая при выполнении запросов к приложения действует как реальный пользователь
2. Время отклика – время, затраченное системой на выполнение запроса пользователя
3. Запросы в секунду – измерение количества запросов отправляемых на сервер



9 Этапов тестирования производительности

1 этап: Анализ системы и рассмотрение требований.

Критерии для измеряемых метрик:

4. Транзакции в секунду – измерение количества транзакций, отправляемых на целевой сервер
5. Процент ошибок – процент ошибок от общего числа ответов в единицу времени
6. Процессор – процентное соотношение времени, которое процессор тратит на выполнение рабочего потока



9 Этапов тестирования производительности

1 этап: Анализ системы и рассмотрение требований.

Критерии для измеряемых метрик:


7. Оперативная память – количество МВ в физической памяти, свободное для всех программ и процессов.
8. Жесткий диск – информация о процентах использования дискового пространства

9 Этапов тестирования производительности

2 этап: Подготовка стратегии.

Стратегия – разрабатывается на основе детального анализа ПО и описывает подходы к тестированию производительности.

Данная стратегия включает всю собранную информацию на этапе анализа системы: требования к ПО, конфигурацию тестового стенда, мониторинг, сценарии поведения пользователей, профили и модели нагрузки, инструментах тестирования производительности, планируемых запусках тестов, виде предоставления результатов



9 Этапов тестирования производительности

2 этап: Подготовка стратегии.

Конфигурация тестового стенда: факторы влияющие на результаты нагрузочного тестирования, загруженность сети, наполненность БД.

Что бы получить достоверные результаты тестирование необходимо проводить на отдельном окружении с параметрами и конфигурацией, которые максимально приближены к характеристикам реальной системы.



9 Этапов тестирования производительности

2 этап: Подготовка стратегии.

Конфигурация тестового стенда: факторы влияющие на результаты нагрузочного тестирования, загруженность сети, наполненность БД.

Что бы получить достоверные результаты тестирование необходимо проводить на отдельном окружении с параметрами и конфигурацией, которые максимально приближены к характеристикам реальной системы.

9 Этапов тестирования производительности

2 этап: Подготовка стратегии.

Структурная схема окружения для тестирования производительности



Сервер БД



Сервер
Приложения



Генератор
нагрузки



9 Этапов тестирования производительности

2 этап: Подготовка стратегии.

Во время тестирования собирается статистика использования ПО, для создания профиля нагрузки – модели поведения пользователей.

Подобный профиль отражает процентное распределение совершаемых в системе операция между пользователями разных ролей. После этого вычисляется начальная точка и размер шага для увеличения интенсивности выполненных операций.

Модель нагрузки может быть разной. Можно добавлять по одному пользователю в минуту, а можно добавить всех сразу



9 Этапов тестирования производительности

3 этап: Настройка генератора нагрузки.

Для качественного проведения тестирования необходимо установить инструмент тестирования на генератор нагрузки – виртуальную или физическую машину, расположенную максимально близко к серверу приложения. Это снижает искажение при подаче нагрузки, вызванные задержками сети.

Для создания большого объема нагрузки ресурсов одной машины может быть недостаточно, по этому необходимо проводить распределенное нагрузочное тестирование

9 Этапов тестирования производительности

3 этап: Настройка генератора нагрузки.



Наша цель – протестировать производительность системы, а не сети.

Правильный подход!

Сеть между тестовым окружением и генератором нагрузки



Тестовое окружение и генератор расположены в одной локальной сети.

Генератор нагрузки



В таком случае мы избежим проблем с производительностью сети и будем уверены в том, что время отклика соответствует реальной производительности приложения.



9 Этапов тестирования производительности

4 этап: Мониторинг серверов и генератора нагрузки.

Выделенные на первом этапе метрики используются для оценки параметров производительности. Информация, полученная при мониторинге всех серверов и ресурсов системы, используется при анализе результатов тестирования.

Мониторинг можно проводить с помощью специальных утилит для отслеживания потребления аппаратных ресурсов ПО.



9 Этапов тестирования производительности

4 этап: Мониторинг серверов и генератора нагрузки.

Задание №1: привести сравнительную характеристику следующих утилит для отслеживания потребления аппаратных ресурсов ПО.

Nmon

Perfomance Monitor

Zabbix

Grafana

New Relic



9 Этапов тестирования производительности

4 этап: Мониторинг серверов и генератора нагрузки.

Задание №2: привести сравнительную характеристику двух инструментов для отслеживания неполадок в БД.

PostgreSQL

SQL Server Profiler



9 Этапов тестирования производительности

5 этап: Подготовка тестовых данных

Универсального подхода к подготовке не существует. Часто для корректной работы решения нежно создавать данные в объеме, достаточном для разработки скриптов и проведения тестов.

Существует 4 способа подготовки данных:

1. Код
2. SQL-запросы
3. API-запросы
4. Интерфейс



9 Этапов тестирования производительности

6 этап: Разработка нагрузочных скриптов.

Сценарии тестирования производительности создаются с использованием выбранного инструмента.

Процесс разработки скриптов состоит из трех шагов:

1. Изучение сценарием – анализ подготовленных сценариев, для определения действия каждого.
2. Создание тест-кейсов – каждый сценарий записывается в шаблон без параметризации.
3. Отладка сценариев – запуск каждого сценария



9 Этапов тестирования производительности

7 этап: Предварительные запуски тестов.

Данный этап необходим для проверки корректности работы подготовленных нагрузочных скриптов и поиска оптимальной модели нагрузки на систему.

9 Этапов тестирования производительности

8 этап: Тестирование.

1. Стресс-тест происходит с постепенно увеличивающейся нагрузки на сервер и возрастает до тех пор пока не будет достигнут один из сценариев остановки теста:

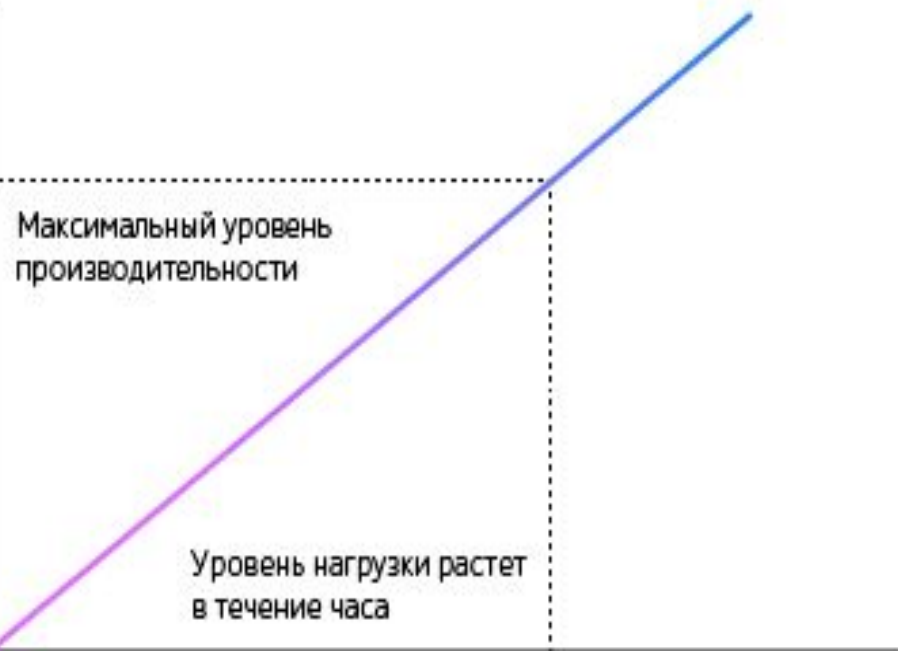
- превышение требуемых значений отклика в несколько раз;
- достижение критического уровня использования аппаратных ресурсов (ЦП>80%, память>90%);
- количество ошибок HTTP превышает 1% от общего числа запросов;
- сбой системного ПО.

9 Этапов тестирования производительности

8 этап: Тестирование.

1. Стресс-тест

Виртуальные пользователи



Продолжительность
теста



9 Этапов тестирования производительности

8 этап: Тестирование.

2. Нагрузочное тестирование – происходит в течении длительного промежутка времени (около 10ти часов).

Если в результате стресс-теста системы не выдержали целевую нагрузку, то проверка проходит под нагрузкой 80% от результата максимальной производительности, полученной при проведении стресс-теста.

9 Этапов тестирования производительности

8 этап: Тестирование.

2. Нагрузочное тестирование

Виртуальные пользователи


Требуемый уровень нагрузки

Продолжительность нагрузки - 4-8 часов

Время нарастания нагрузки

Продолжительность теста





9 Этапов тестирования производительности

8 этап: Тестирование.

3. Проверка стабильности проводится с ожидаемым уровнем нагрузки при длительном тестировании. При достижении максимального количества пользователей нагрузка на сервер больше не подается и остается постоянной на протяжении всей проверки. Может продолжаться в течении нескольких дней.

9 Этапов тестирования производительности

8 этап: Тестирование.

3. Проверка стабильности

Виртуальные пользователи



Продолжительность нагрузки ~ 7 дней

Продолжительность теста



9 Этапов тестирования производительности

8 этап: Тестирование.

4. Объемное тестирование используется для оценки производительности работы ПО при увеличении данных, которые хранятся в БД приложения. Для проведения такого теста необходимо заполнить базу определенным объемом информации

9 Этапов тестирования производительности

8 этап: Тестирование.

3. Объемное тестирование

Такая же форма нагрузки, как и для нагрузочного теста





9 Этапов тестирования производительности

8 этап: Тестирование.

5. Проверка масштабируемости позволяет оценить способность решения увеличивать производительность пропорционального добавлению аппаратных ресурсов системы

9 Этапов тестирования производительности

8 этап: Тестирование.

3. Проверка масштабируемости





9 Этапов тестирования производительности

8 этап: Тестирование.

6. При модульном тестировании нагрузка подается на отдельные компоненты ПО

7. Тест «часа пик» проводится для того, что бы проверить реакцию системы как в момент наибольшей загруженности, так и непосредственно после снижения нагрузки

8. Тестирование конфигурации оценивает работу программного продукты при разных конфигурациях окружения



9 Этапов тестирования производительности

9 этап: Анализ результата и подготовка отчета

На основе предыдущих этапов создается документ, который описывает не только результаты проведенных тестов, но и ход выполнения каждого из них

9 Этапов тестирования производительности

9 этап: Анализ результата и подготовка отчета

Отчёт по тестированию содержит следующую информацию:

- **цель проведения тестирования;**
- **конфигурацию тестового стенда и генератора нагрузки;**
- **требования к ПО;**
- **сценарии поведения пользователей, профиль нагрузки;**
- **статистику по ключевым характеристикам производительности (время отклика, количество запросов в секунду, количество транзакций в секунду);**
- **данные о максимально возможном количестве одновременно работающих пользователей, при котором решение будет справляться с нагрузкой;**
- **сведения о количестве и типах ошибок HTTP;**
- **графики, показывающие зависимость производительности системы от количества одновременно работающих пользователей;**
- **выводы о производительности системы в целом и о слабых местах, если они будут обнаружены;**
- **рекомендации по улучшению производительности ПО.**