

# Нагрузочное тестирование vs Тестирование производительности

## Нагрузочное тестирование

Сегодня мы немного поговорим про теорию тестирования. Очень часто можно услышать вопрос: “Как же правильно говорить: **Нагрузочное тестирование** или Тестирование производительности? И чем одно от другого отличается?”. В русскоязычной среде термины “**Нагрузочное тестирование**” и “Тестирование производительности” перепутаны, и не всегда понятно откуда что взялось.

## Введение

- 1 Software testing consists of the dynamic verification that a program provides
- 2 expected behaviors on a finite set of test cases, suitably elected from the usually infinite ex
- 3 IEEE Guide to Software Engineering Body of Knowledge, SWEBOK V3.0, 2014]

Или по-русски:

- 1 Тестирование программного обеспечения заключается в динамической проверке того,
- 2 что программа обеспечивает ожидаемое поведение на ограниченном наборе тестов,
- 3 обычно выбранных из бесконечного списка тестов определенным образом.

На текущий момент существует множество видов тестирования также существует большое количество классификаций эти видов. Основная классификация видов тестирования происходит по целям. На рисунке ниже представленная классификация видов тестирования.



Классификацию тестирования можно продолжать и дальше, разделяя на ещё более атомарные единицы данные типы тестирования. Основная сложность при классификации видов тестирования заключается в том, что в основном все стандарты существуют на английском языке, и при переводе терминов могут возникать коллизии взаимозависимости от того, как тот или иной термин будет переведен на русский язык. В связи с этим, кроме русского перевода вида тестирования в скобках будет указываться его английское наименование. В рамках данной статьи мы рассмотрим более подробно тестирование производительности (Performance Testing).

# Тестирование производительности (Performance Testing)

Считается, что тестирование производительности [1] — это то тестирование, которое не является функциональным. Существует множество видов тестирования производительности. Классификация видов тестирования производительности строится на основе того, какие цели преследует определенный вид тестирования. Как правило тестирование производительности преследует не одну, а несколько целей в связи с тем, многие типы тестирования в ходе его проведения совмещаются с другими целями или повторяются несколько раз в ходе цикла тестирования. Основное отличие тестирования производительности также заключается в том, что оно происходит только после полного функционального тестирования. Ошибки функциональности не исправляются в ходе тестирования производительности. Для данного вида тестирования чаще всего выделяется отдельный нагрузочный стенд, повторяющий копию промышленного стенда. В связи с массовым распространением Agile методологий тестирование производительности также интегрируется в жизненный цикл разработки программного обеспечения.

На рисунке ниже показана основная классификация видов тестирования производительности.



На основе классификации показанной на рисунке выше можно выделить основные виды тестирования, которые производятся во время этапа тестирования производительности системы. Основные типы тестирования и вопросы, которые они решают представлены в таблице.

№	Вид тестирования	Вид тестирования по английский	Вопрос на который отвечает тестирование
1	Нагрузочное тестирование	Load Testing[2]	Достаточно ли быстро работает система?
2	Тестирование стабильности	Stability Testing[3]	Достаточно ли надежно работает система на долгом интервале времени?
3	Тестирование отказоустойчивости	Failover Testing[4]	Сможет ли система переместиться сама на другой сервер в случае сбоя основного сервера?
4	Тестирование восстановления	Recovery Testing[5]	Как быстро восстановится система?
5	Стрессовое тестирование	Stress Testing[6]	Что произойдет при незапланированной нагрузке?
6	Тестирование объемов	Volume Testing[7]	Как будет работать система, если объем базы данных увеличится в 100 раз?
7	Тестирование масштабируемости	Scalability Testing[8]	Как будет увеличиться нагрузка на компоненты системы при увеличении числа пользователей?
8	Тестирование потенциальных возможностей	Capacity Testing[9]	Какое количество пользователей может работать?
9	Конфигурационное тестирование	Configuration Testing[10]	Как заставить систему работать быстрее?
10	Тестирование сравнения	Compare Testing[11]	Какое оборудование и ПО выбрать?

## Нагрузочное тестирование (load testing)

Нагрузочное тестирование (load testing) – данный тип тестирования позволяет оценить поведение системы при возрастающей нагрузке, целью нагрузочного тестирования является также определение максимальной нагрузки, которую может выдержать система.

Это наиболее распространенный и известный тип тестирования, и поэтому в русскоязычной среде он на слуху. Часто в рунете, особенно те, кто не в теме QA, под нагрузочным тестированием понимают все виды испытаний. Но, в англоязычной литературе, это всего лишь подвид тестирования производительности.

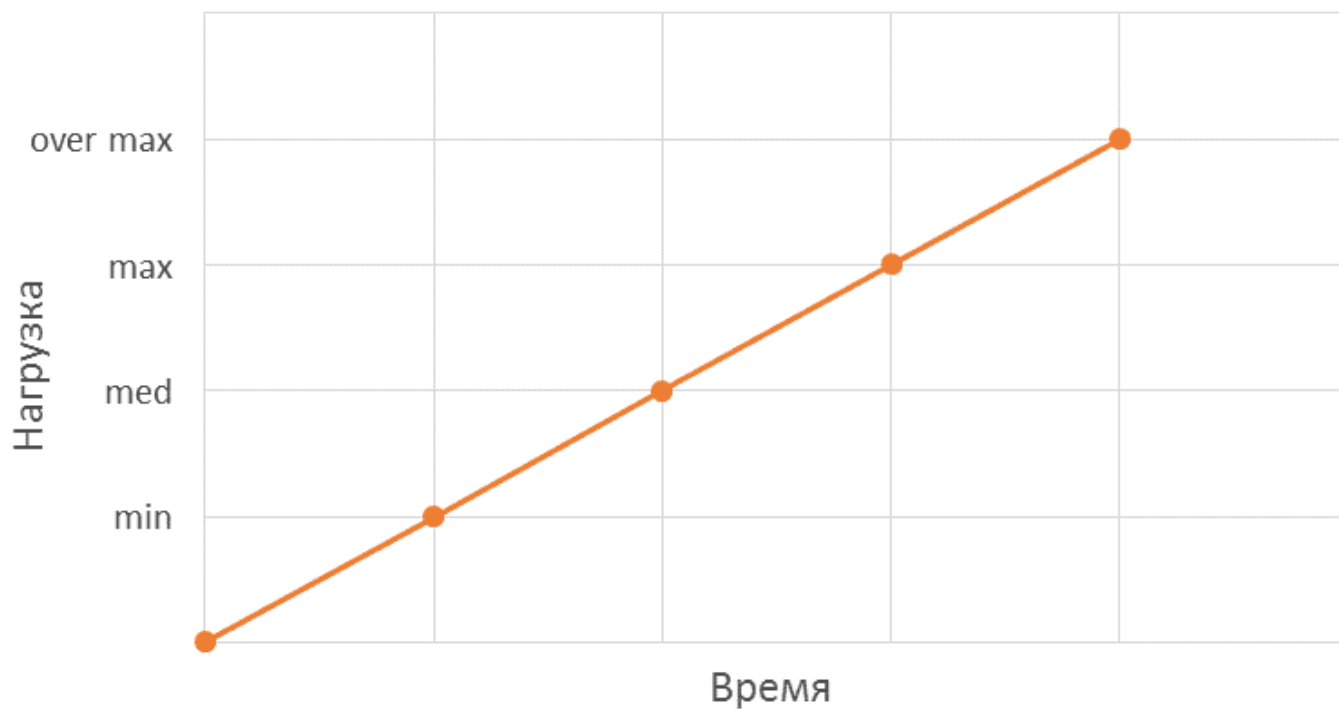
Рассмотрим его подробнее: В роли нагрузки может выступать количество пользователей, а также количество операций на сервере.

Производительность при этом определяется следующими факторами:

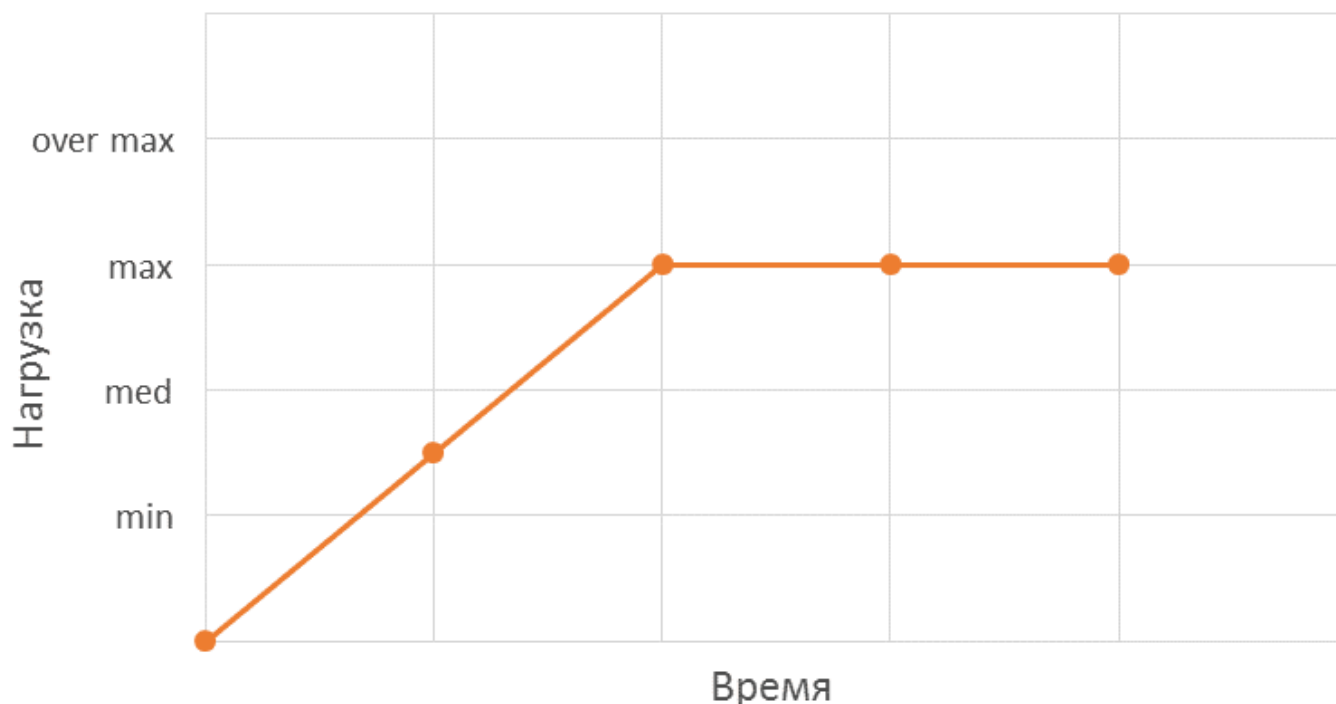
- скоростью работы программного обеспечения;
- скоростью работы аппаратного обеспечения;
- скоростью работы сети.

Во время тестирования могут осуществляться следующие операции, позволяющие более точно измерить производительность и определить “узкое место” системы [12]:

- измерение времени выполнения выбранных операций при определенных интенсивностях выполнения этих операций;
- определение количества пользователей, одновременно работающих с приложением;
- определение границ приемлемой производительности при увеличении нагрузки (при увеличении интенсивности выполнения этих операций).



После нахождения максимальной производительности рекомендуется её “подтвердить”. Для этого проводится дополнительный тест со следующим профилем:

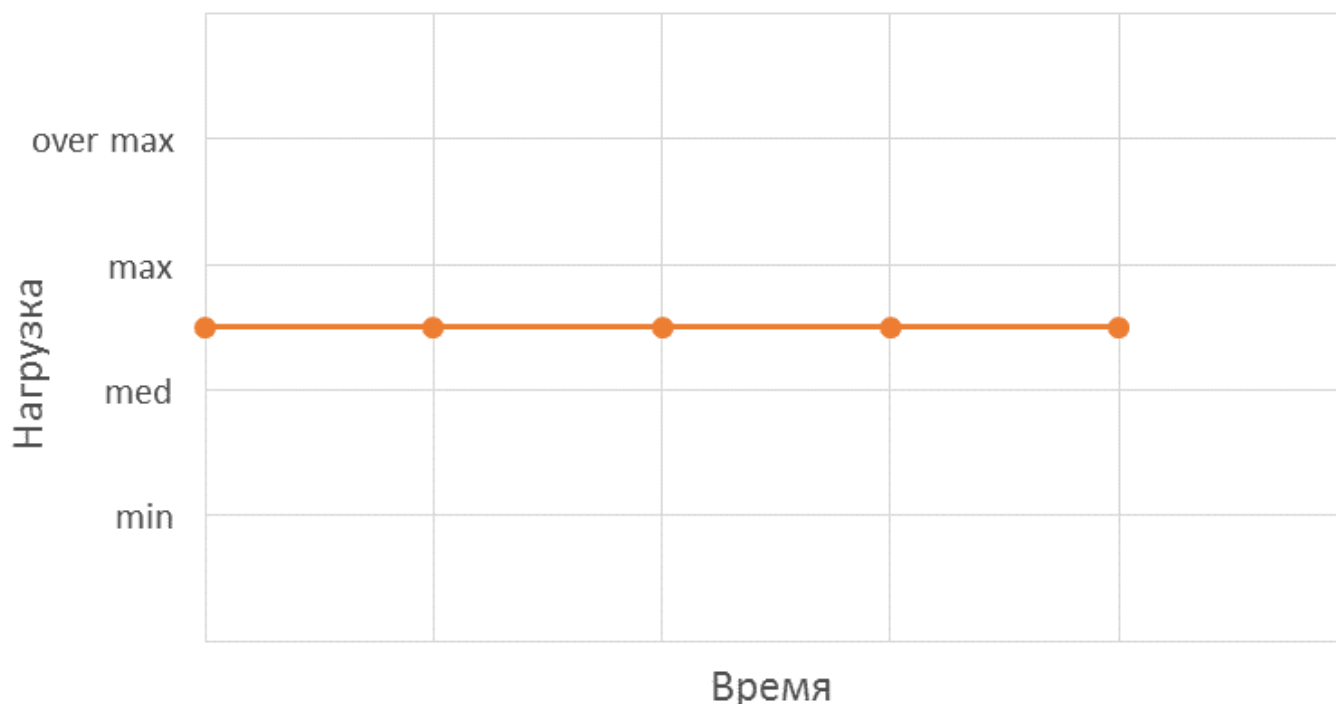


## Тестирование стабильности (stability testing)

Тестирование стабильности (stability testing) — позволяет проверить работоспособность системы на длительном интервале времени. При этом нагрузка может не достигать пиковых значений, а иметь среднее значение, так же само время выполнения операций не является основным фактором в оценке результатов тестирования.

В ходе тестирования основной акцент делается на измерение

- отсутствие утечек памяти;
- отсутствие перезапусков серверов;
- отсутствие перезапусков программного обеспечения;
- любые ошибки, связанные с накоплением данных;
- отсутствие отключений или сбоев в работе сетевого оборудования.



## Тестирование отказоустойчивости (failover testing)

Тестирование отказоустойчивости (failover testing) – данный вид тестирования производительности позволяет проверить поведение системы в случае сбоя серверов или при других неблагоприятных факторах. Такое тестирование особенно важно в системах, работающих в режиме 24/7, т.к. в случае их выхода из строя возможны потери клиентов, репутации, денег и т.п.

Во время тестирования проверяются следующие операции:

- как будет преодолеваться отказ, а именно как система будет перемещать операции между мощностями работающего и нет оборудования;
- как будет осуществлен перехват управления системой при отказе управляющего сервера;
- как будет осуществлен обход и обработка отказа (переключение на резервный канал связи, отправка данных по другому маршруту и т.д.).

## Тестирование восстановления (recovery testing)

Тестирование восстановления (recovery testing) – обычно неотъемлемо связан с тестированием отказоустойчивости и позволяет определить, как быстро система сможет восстановиться после сбоя её программной или аппаратной части. Сам сбой производится тестировщиком путём отключения, например, одного из серверов или его перезагрузки. Тестирование не нацелено на проверку надёжности системы. При этом нагрузка на систему не уменьшается и имеет средние или пороговое значение.

В ходе тестирования измеряются следующие показатели:

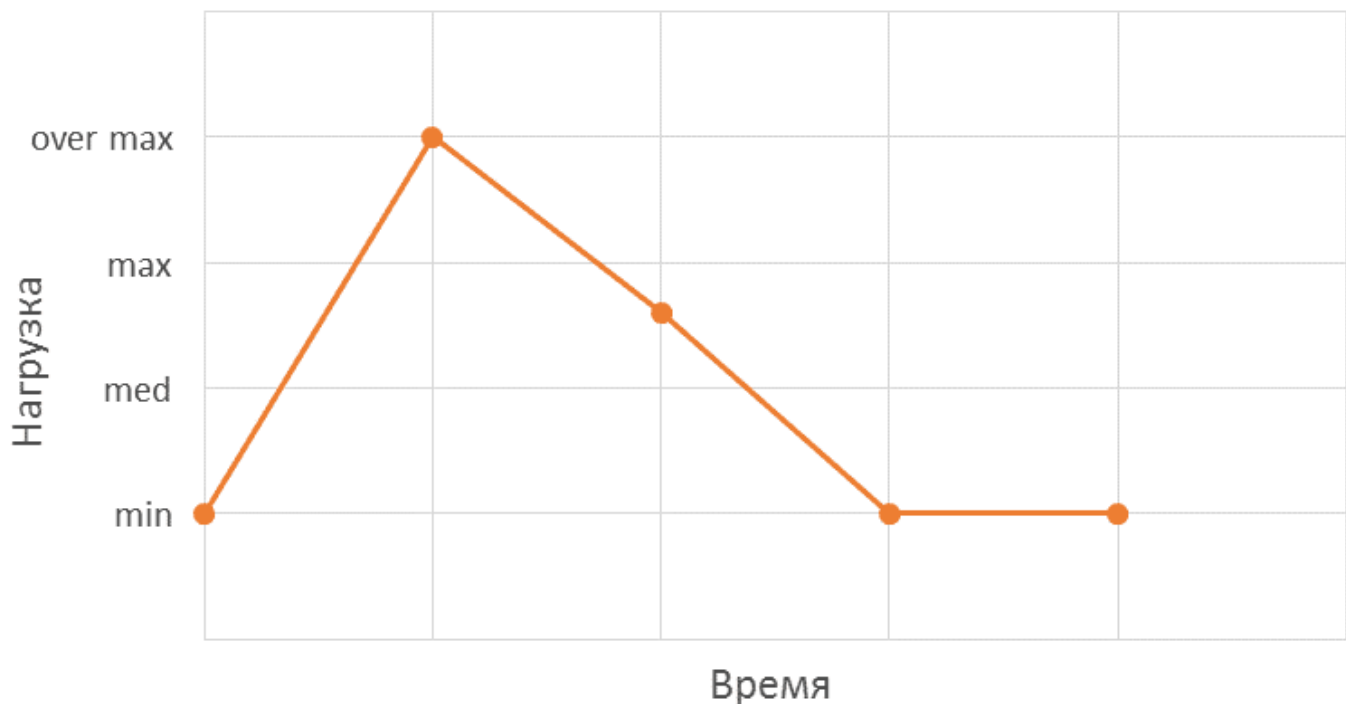
- время, за которое система восстановится после сбоя;
- корректность восстановленных данных.

## Стрессовое тестирование (stress testing)

Стрессовое тестирование (stress testing) — целью данного вида тестирования производительности является оценка производительности системы при пороговых значениях рабочей нагрузки или за её пределом. Также в ходе тестирования можно оценивать работу системы при изменении ресурсов доступных системе таких как процессорное время, память, ширина сетевого канала и т.д.

В ходе тестирования измеряется:

- возможность и время регенерации системы – возможность и время возвращения системы к нормальному состоянию после стрессовых нагрузок;
- корректность логирования ошибок и оповещений о их возникновении;
- производительность системы при стрессовой нагрузке;
- оценка влияния сбоев тестируемой системы на внешние системы.



## Тестирование объемов (volume testing)

Объемное тестирование (volume testing) — тестирование позволяет оценить производительность системы при увеличении объемов данных как самого приложения, так и его базы данных. Основной вопрос, на который отвечает данный вид тестирования производительности: “Что будет завтра с этим приложением или через год при увеличении числа пользователей и/или увеличение хранимых пользовательский и системных данных?”.

Во время тестирования измеряются следующие параметры:

- зависимость времени выполнения операций на сервере от объема данных;
- количество пользователей, которые могут одновременно работать с приложением “быстро”;
- как быстро увеличивается объем данных при работе приложения.

## Тестирование масштабируемости (scalability testing)

Тестирование масштабируемости (scalability testing)[13] – данное тестирование производится для проверки возможностей масштабирования приложения под любым видом нагрузки. Также необходимо проверять производительность системы во время масштабирования.

Виды масштабирования, которые проверяются в ходе тестирования:

- вертикальное масштабирование – увеличения производительности каждого отдельного компонента системы (добавление оперативной памяти на сервере, замена процессора и т.д.) для повышения производительности всей системы в целом;
- горизонтальное масштабирование – распределение системы на большее количество серверов параллельно работающих и выполняющих одни и те же функции;
- применение временного масштабирования внутри системы с помощью очередей, асинхронных запросов и т.п.

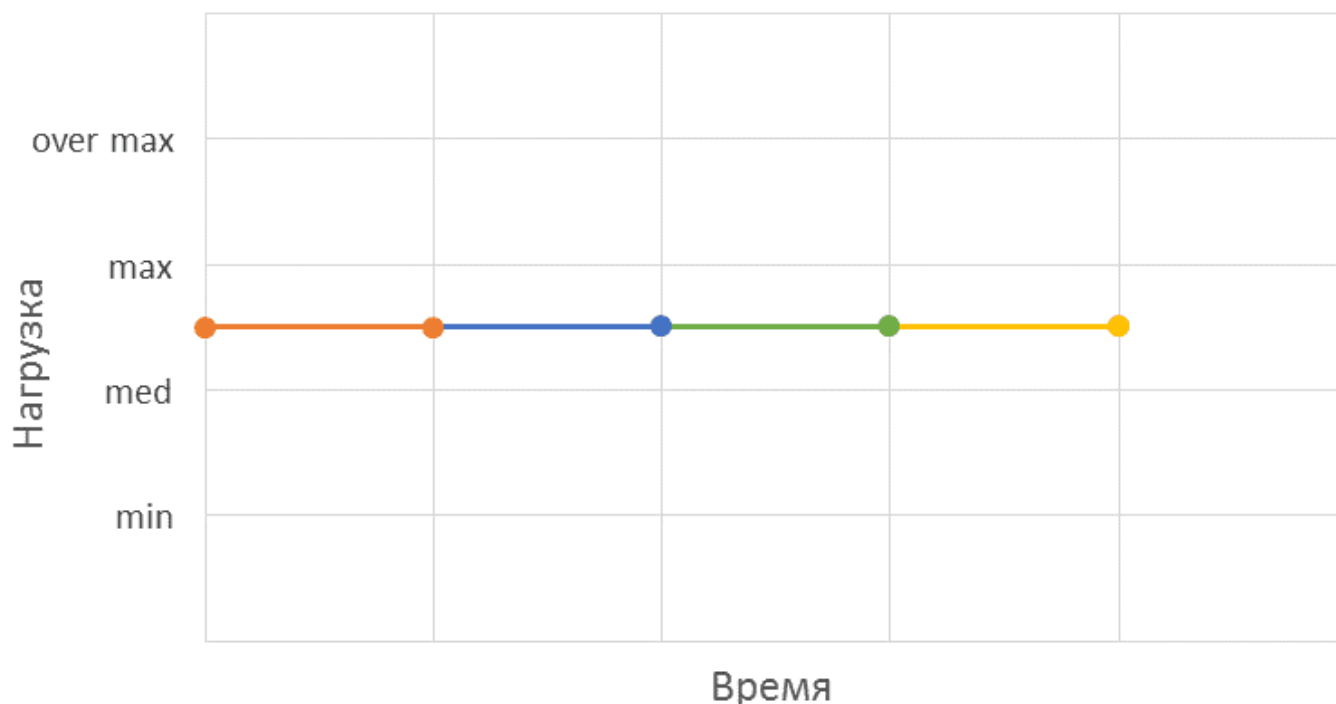
## Тестирование потенциальных возможностей (capacity testing)

Тестирование потенциальных возможностей (capacity testing) – является отчасти подвидом тестирования масштабируемости. Если в тестирование масштабируемости основной вопрос, который ставится: насколько хорошо справляется система с возрастающим количеством пользователей (нагрузки). То в данном случае вопрос звучит следующим образом: сколько пользователей (с какой нагрузкой) может работать с системой при этом время отклика и другие параметры производительности должны находится в пределах допустимых значений? Данный вид тестирования позволяет определить стратегию масштабирования и вообще понять, а стоит ли масштабировать систему?



# Конфигурационное тестирование (configuration testing)

Конфигурационное тестирование (configuration testing) [14] – данный вид тестирования проверяет производительность системы на разных аппаратных и программных конфигурациях. В ходе тестирования измеряются основные показатели производительности системы при средних и пороговых значениях нагрузки. Данное вид тестирования производительности позволяет убедиться, что на других конфигурациях аппаратного и программного обеспечения система будет работать с одинаковой производительностью.



## Тестирования сравнения (compare testing)

Тестирования сравнения (compare testing) – позволяет сравнить производительности на разной конфигурации программной и аппаратной части системы. Данное тестирование помогает выбирать наиболее оптимальную конфигурацию аппаратного и программного обеспечения. В ходе тестирования производится проверка на разных конфигурациях, при этом профиль тестирования не изменяется от конфигурации к конфигурации и имеет среднюю или пороговую интенсивность нагрузки.

Тестирование позволяет ответить на такие вопросы как:

- какую СУБД выбрать?
- какое оборудование выбрать (платформа, производитель, цена и т.д.)?

- как повлияют на работу приложения обновления и патчи?

## Выводы

Ну хорошо, из этой статьи мы видим что видов тестирования Производительности много. Очень много, даже слишком много. А какой нужно использовать для конкретно нашей системы? А почему? А сколько делать по времени тесты стабильности? А нужно ли проводить тестирование изолированно или в комплексе с интеграцией? А что делать если нет стенда, сил, времени на такое тестирование? Обращайтесь к нам, в Перфоманс Лаб, мы с удовольствием вам поможем.

Кстати, у нас скоро будет конференция по нагрузочному тестированию, где мы расскажем не только про теорию, но и про практику и про самые последние новости в этой области.