Тема лекции

Тестирование производительности и нагрузочное тестирование

При подготовке лекции использованы материалы доцента кафедры ПОИТ С.С. Куликова

Вопросы

1. Основные понятия.
2. Виды тестирования производительности и решаемые задачи
3. Этапы тестирования

ОПРЕДЕЛЕНИЯ ХАРАКТЕРИСТИК КАЧЕСТВА (ISO 9126-1)

- ***Эффективность*** - свойства программного средства, обеспечивающие требуемую производительность решения функциональных задач, с учётом количества используемых вычислительных ресурсов в установленных условиях.

Основные понятия

- **Виртуальный пользователь (virtual user)** – процесс (программа), выполняющий некоторые заданные операции.
- **Итерация (iteration)** – повтор операции.
- **Интенсивность выполнения операции (operation intensity)** – количество итераций в единицу времени.
- **Нагрузка (load, loading)** – совокупность выполняемых с системой операций.
- **Производительность (performance**) – количество операций, выполняемых в единицу времени.
- **Масштабируемость (scalability)** – способность системы увеличивать производительность пропорционально добавлению ресурсов. .

Основные понятия

- Тестирование производительности (performance testing) – исследование «скоростных показателей» приложения при различной по характеру и количественных показателях нагрузке.
- Нагрузочное тестирование (load testing) – исследование способности приложения сохранять заданные показатели качества при нагрузке в допустимых пределах и некотором превышении этих пределов (определение «запаса прочности»).
- Нагрузочное тестирование ≠ Тестирование производительности.

Основные понятия

- Стрессовое тестирование (stress testing) –исследование поведения приложения при «экстремальных» изменениях нагрузки.
- Объёмное тестирование (volume testing) – исследование производительности приложения при обработке различных объёмов данных.

Примеры

- Тестирование производительности – с какой скоростью машина может ехать с 1-2-3-4-5-ю пассажирами, в гору, с горы, на разных видах топлива, как быстро она разгоняется и т.п.
- Нагрузочное тестирование – может ли машина при некоторых установленных параметрах загрузки, топлива и т.п. разогнаться до 100 км/ч за указанное время и сохранять такую скорость на протяжении указанного маршрута. А если её перегрузить?
- Стрессовое тестирование – зима, -50, если её завести и с 15-ю пассажирами подниматься гору на 5-й передаче…
- Объёмное тестирование – при каком количестве пассажиров (груза) начнутся проблемы со скоростными характеристиками.

Примеры (принтер)

- Тестирование производительности – за какое время принтер напечатает N страниц вот таких, таких и таких документов.
- Нагрузочное тестирование – если на протяжении пяти часов непрерывно печатать…
- Стрессовое тестирование – бумага со скрепками, кусок картона, 15000 одновременно посланных на печать документов (все с наивысшим приоритетом), печатать непрерывно трое суток…
- Объёмное тестирование – послать на печать маленький txt-документ, 500-страничный doc-документ, 27-гигабайтный psd-документ .

Примеры (сайт)

- Тестирование производительности – каково максимальное время генерации страницы при 10-20-500 пользователях, если их число резко меняется, если они выполняют такие-то действия…
- Нагрузочное тестирование – заявленные в спецификации 100 пользователей одновременно сидят на сайте и выполняют некоторые действия… как ведёт себя сайт? А если пользователей будет 120?
- Стрессовое тестирование – 500-1000-5000 пользователей, DoS/DDoS атака, наплыв спамеров…
- Объёмное тестирование – в базе новостей 10000000 записей, на форуме 1000000 сообщений….

Цели тестирования производительности:

- оценка времени выполнения выбранных операций при определённой интенсивности и очерёдности выполнения этих операций;
- оценка реакции приложения на изменение количества пользователей, одновременно работающих с приложением;
- оценка границ интенсивности и видов нагрузки, при которых производительность приложения выходит за рамки приемлемой;
- исследование производительности на низких, средних, высоких, предельных и стрессовых значениях нагрузки;
- оценка показателей масштабируемости приложения.

Цели нагрузочного тестирования :

- оценка скорости реакции приложения на различные значения нагрузки в допустимых пределах;
- оценка использования приложением системных ресурсов при различных значениях нагрузки в допустимых пределах;
- оценка изменения со временем поведения приложения при сохранении некоторой допустимой нагрузки длительное время.

Цели стрессового тестирования:

- оценка реакции приложения на нестандартные и стрессовые случаи изменения нагрузки, в т.ч.:
  - резкое непредсказуемое изменение интенсивности нагрузки;
  - значительное превышение предельно допустимой нагрузки;
  - интенсивное использование функций приложения, являющихся «узким местом» в производительности

Цели объемного тестирования:

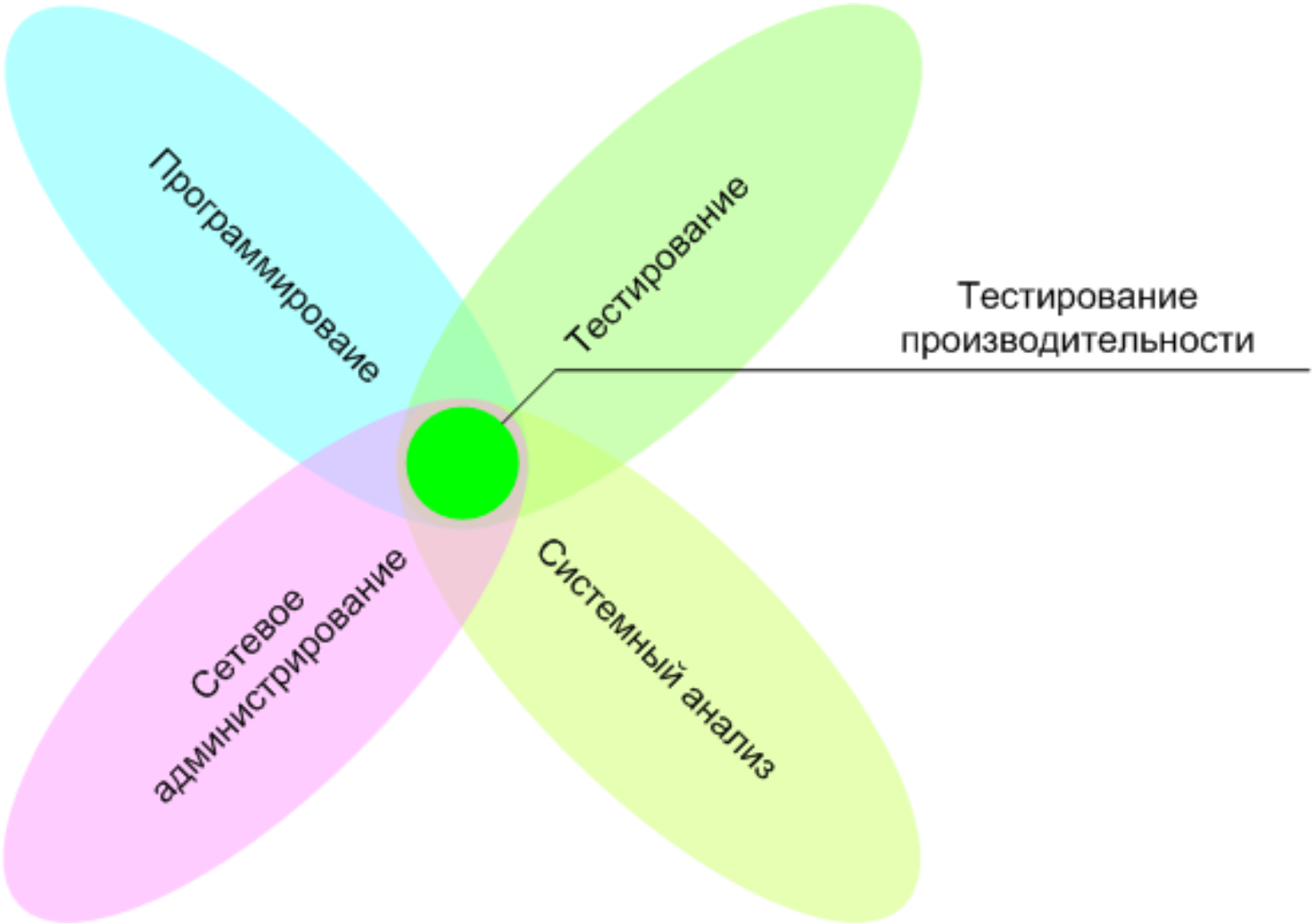
- оценка показателей производительности приложения в случаях приёма, обработки и генерации данных различного объёма и с различными показателями вычислительной сложности обработки;
- оценка способности приложения обрабатывать большие объёмы данных в условиях высокой загрузки системных вычислительных ресурсов;
- оценка способности приложения обрабатывать большие объёмы данных при недостатке оперативной памяти

Цели

- определение лучшей архитектуры системы, выбор наилучшей платформы, средств и языков реализации;
- определение оптимального способа хранения файлов;
- оценка и оптимизация схемы БД в контексте повышения производительности;
- определение узких мест системы;
- оценка максимальной и минимальной производительности системы и условий их достижения;
- определение характера увеличения времени отклика системы при увеличении нагрузки;
- определение максимального числа одновременно работающих пользователей, превышение которого делает использование системы невозможным;
- определение влияния конфигурации системы на производительность;
- оценка показателей масштабируемости системы;
- оценка архитектуры и настройки сети заказчика требованиям производительности.

Причины

- Низкую производительность замечает почти любой пользователь, что приводит к его недовольству.
- Низкая производительность уменьшает количество полезных операций в единицу времени, что приносит убытки заказчику.
- Низкая производительность может стать причиной выхода приложения из строя, проблем с безопасностью и т.п.
- Низкая производительность может быть обусловлена некоторыми скрытыми причинами, влияющими и на остальные показатели качества системы (отказоустойчивости, восстанавливаемости, живучести и т.п.)


Программирование
Тестирование
Тестирование производительности
Сетевое администрирование
Системный анализ

Тестирование производительности: виды и решаемые задачи

Характеристики производительности

- время отклика;
- мощность;
- стабильность;
- масштабируемость.

Характеристики производительности

- время отклика;

Низко-, средне- и высоконагруженная работа (low-, mid-, high-load) – позволяет оценить время отклика (response time and latency) системы

Тест «часа пик» (rush hour test) – позволяет оценить реакцию системы на резкое изменение нагрузки.

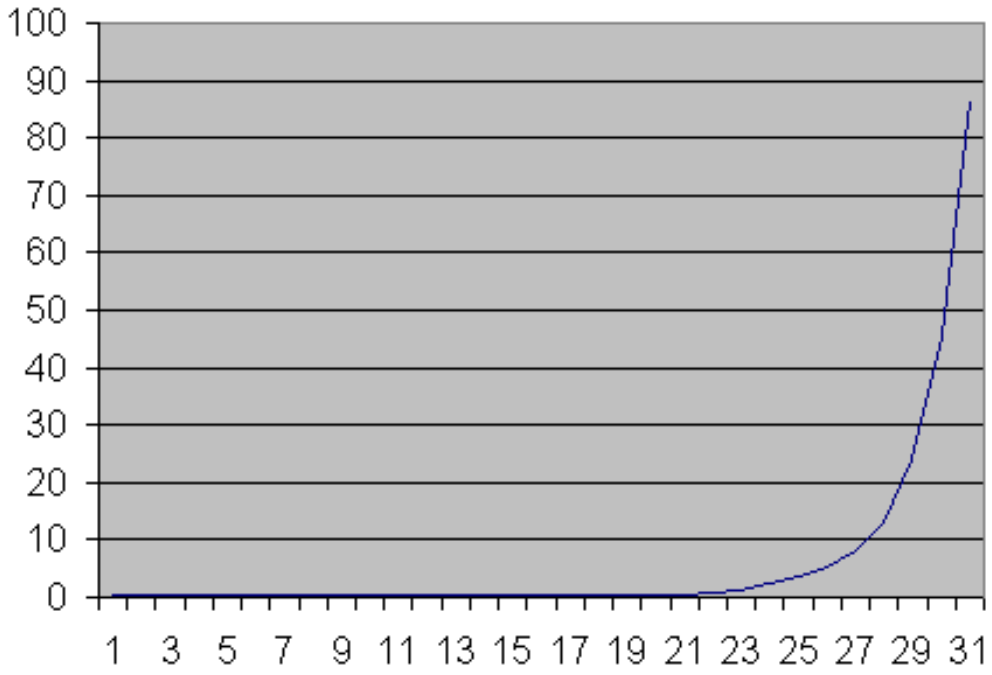
Тест на выживаемость (longevity test) – показывает способность системы работать длительное время под высокой нагрузкой.

Наращивание нагрузки (ramp up) – позволяет определить т.н. «точку насыщения» (saturation point) – показатель нагрузки, при которой производительность системы начинает ухудшаться. Проведение нескольких ramp up тестов с увеличением доступных аппаратных ресурсов позволяет определить, насколько система масштабируема (scalability).

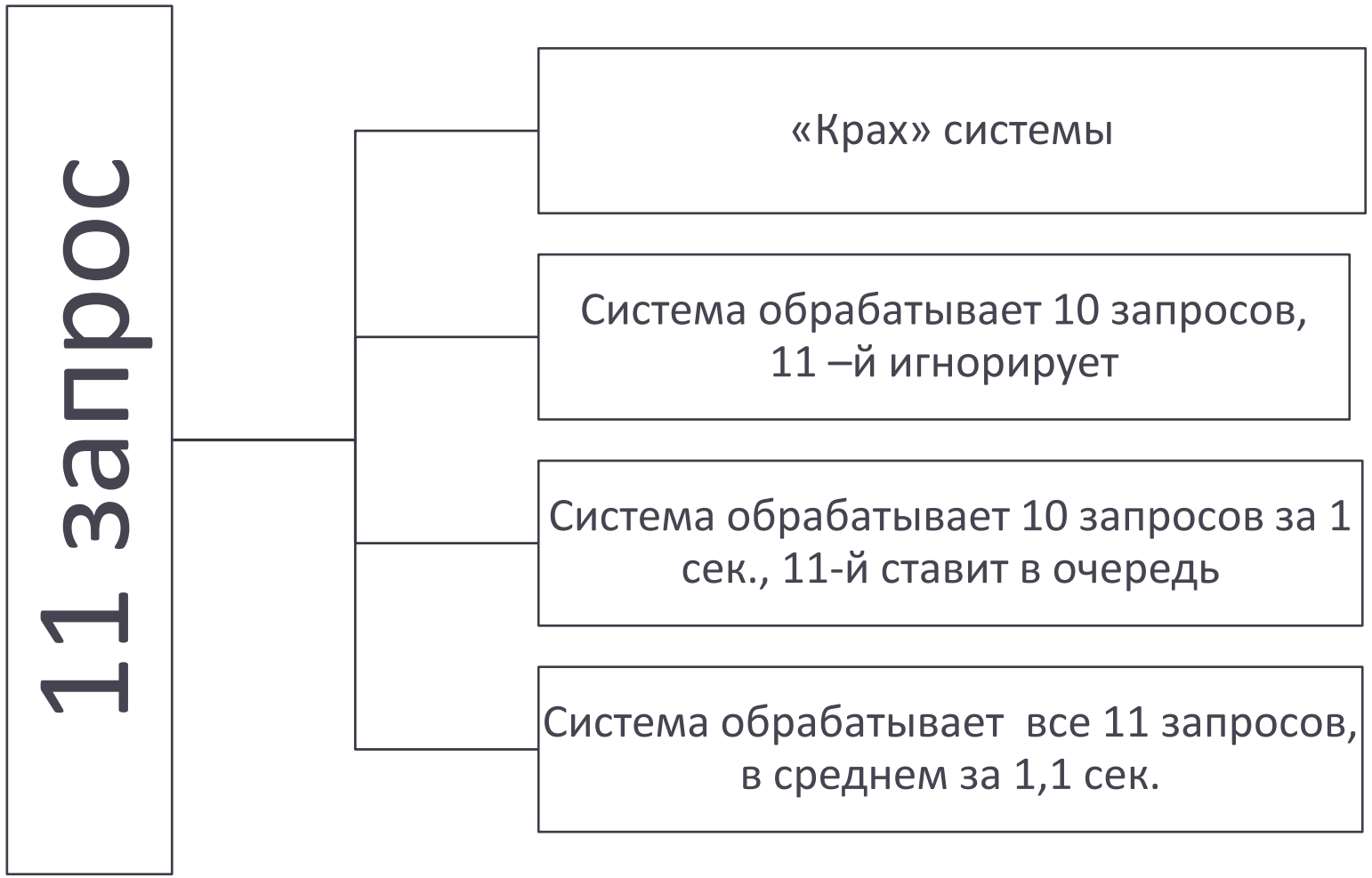
Примеры

Скрипт, генерирующий дерево комментариев к статье, в среднем обрабатывает за 0.5 секунды, если уровень вложенности комментариев не превышает 21. Затем время работы выглядит так:

- 22 уровня = 0.81 секунды
- 23 уровня = 1.12 секунды
- 24 уровня = 2.53 секунды
- 25 уровней = 3.52 секунды
- 26 уровней = 5.47 секунды
- 27 уровней = 8.05 секунды
- 28 уровней = 13.16 секунды
- 29 уровней = 24.15 секунды
- 30 уровней = 45.52 секунды
- 31 уровень = 86.61секунды



Примеры: rush up test



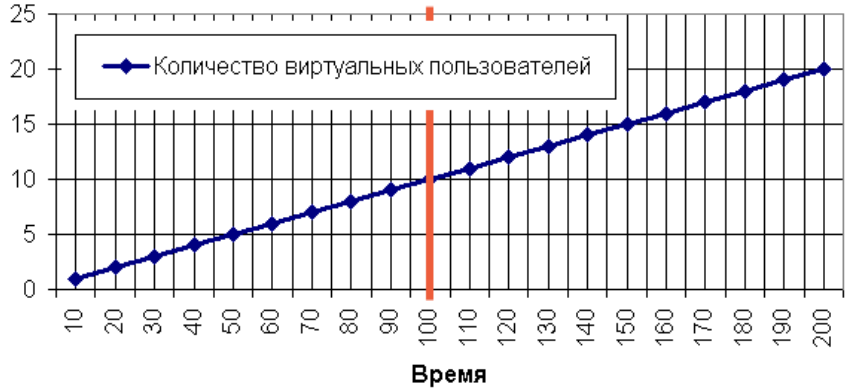
Примеры: rush up test

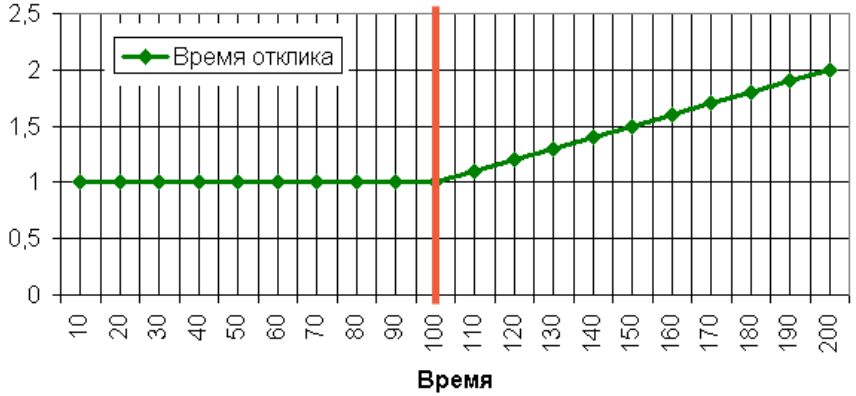
Варианты поведения системы:

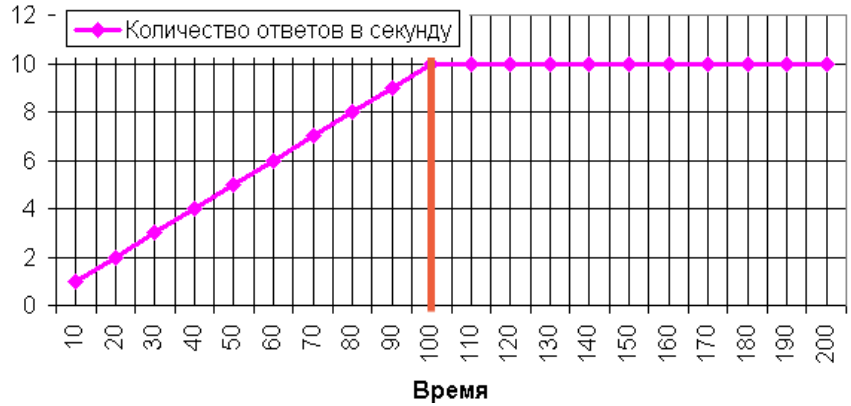
- **хороший**
- **приемлемый**
- **плохой**
- **ужасный**

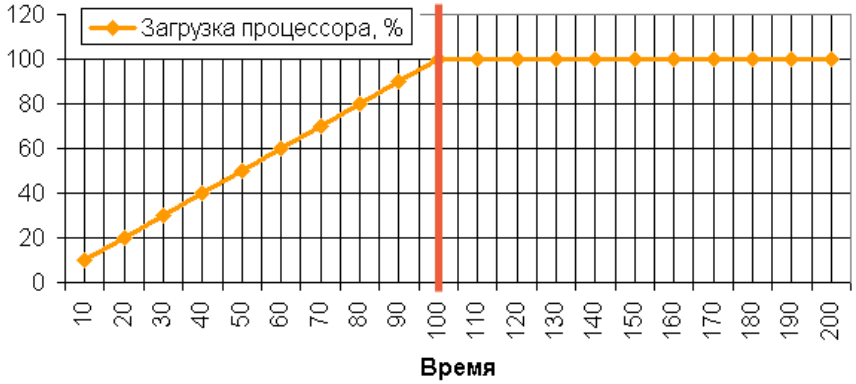
Примеры: rush up test

- Хороший (почти идеальный)



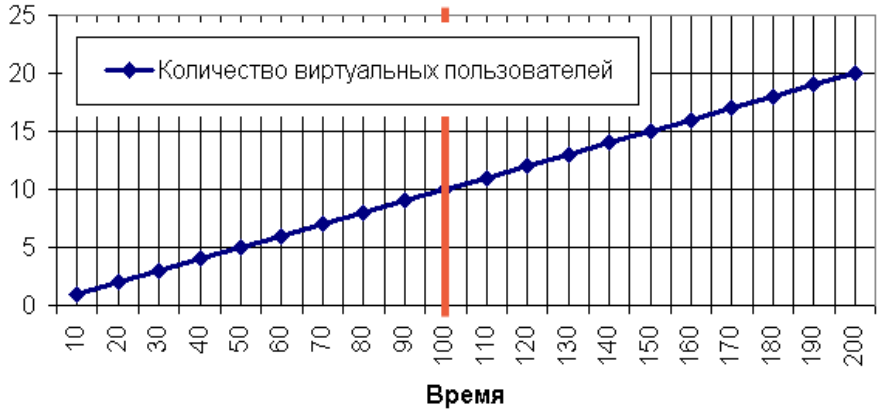


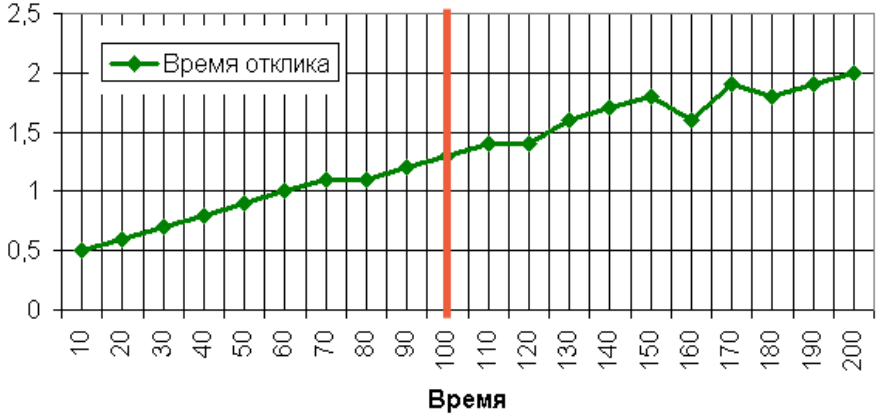


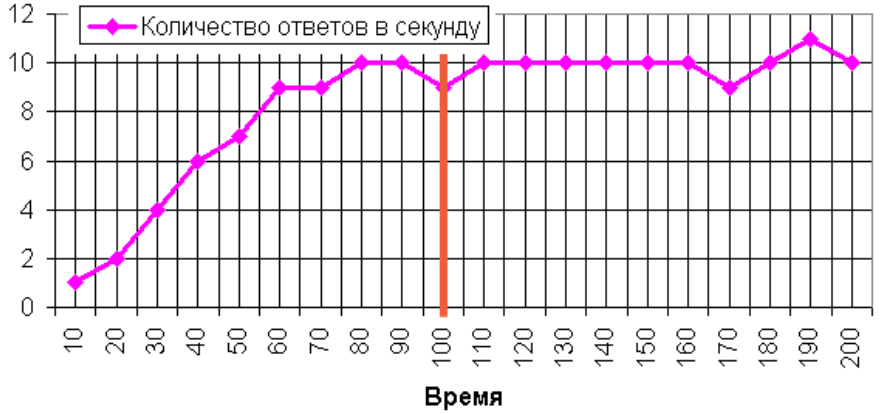


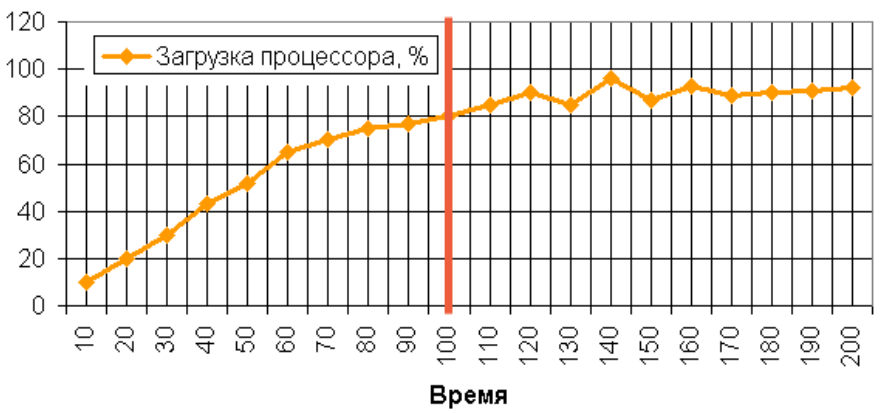
Примеры: rush up test

- Приемлемый



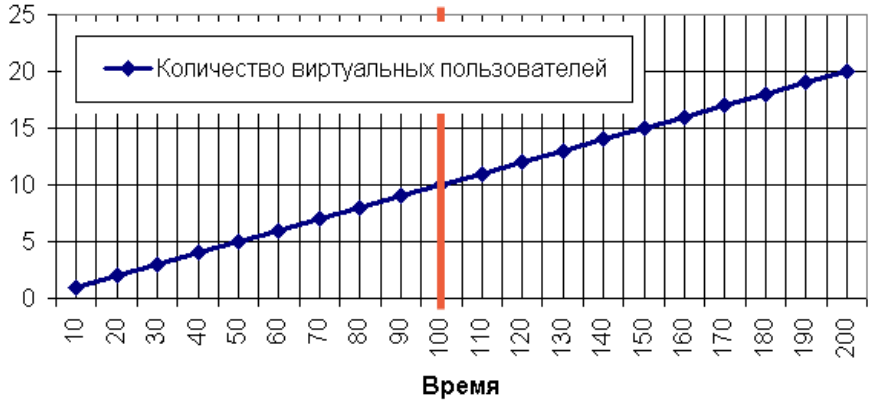


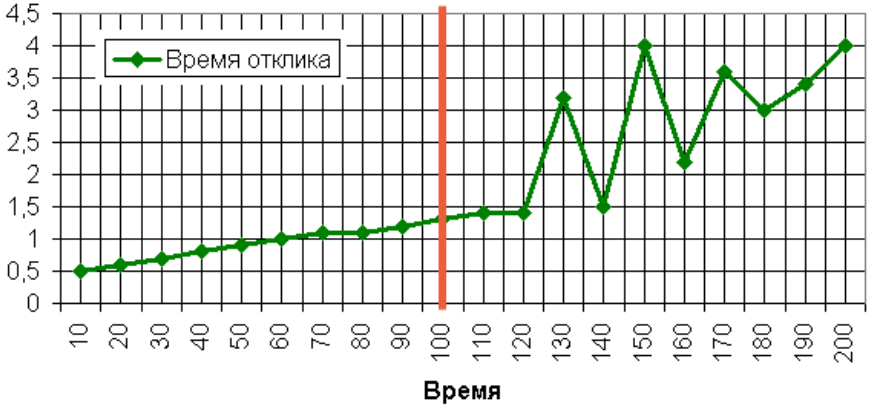


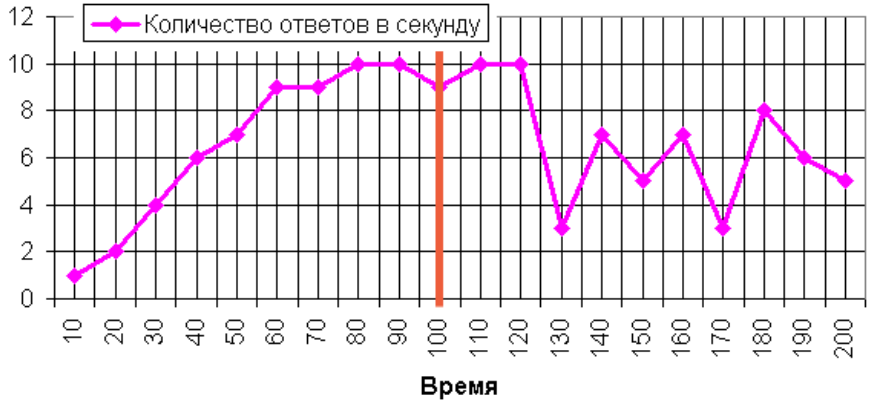


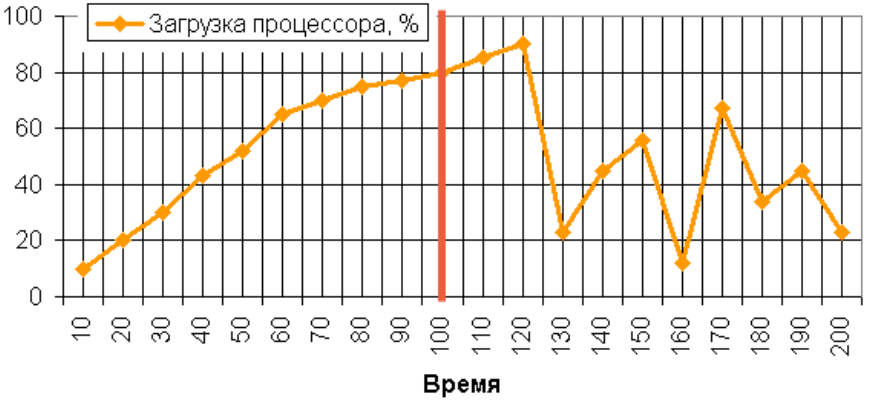
Примеры: rush up test

- Плохой



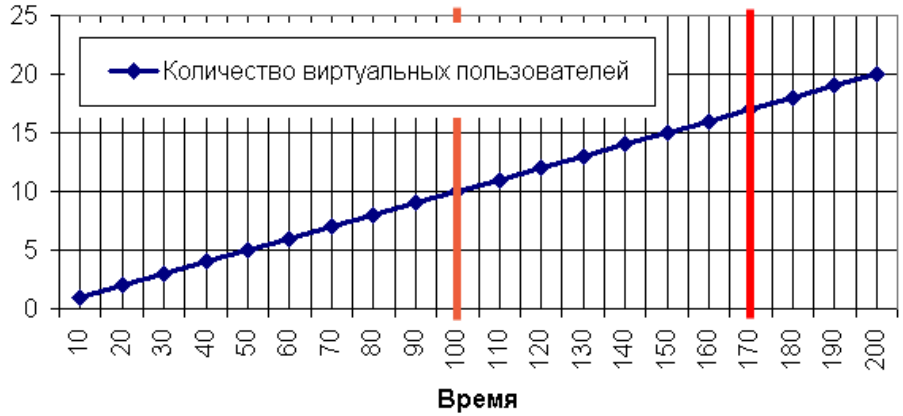


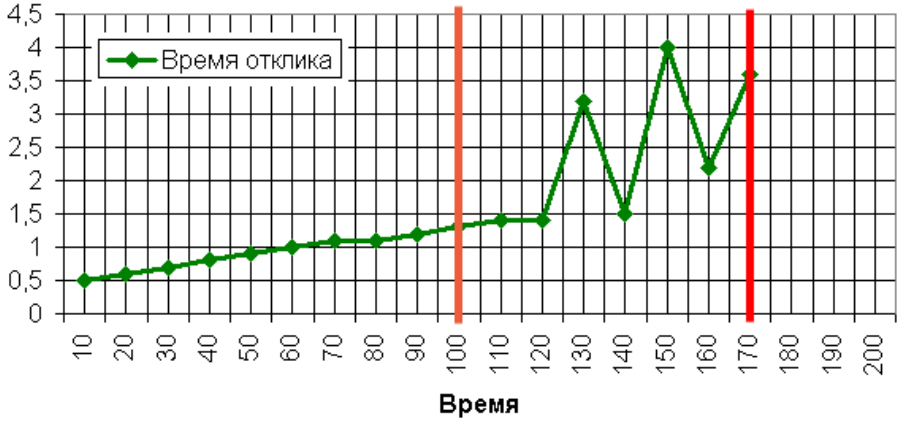


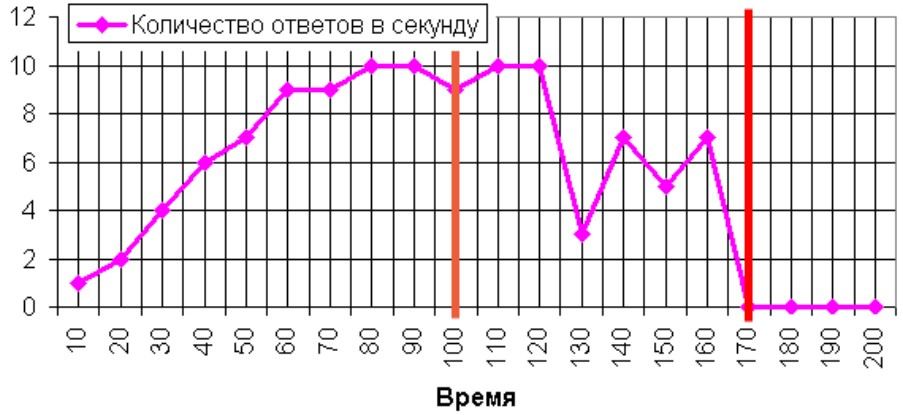


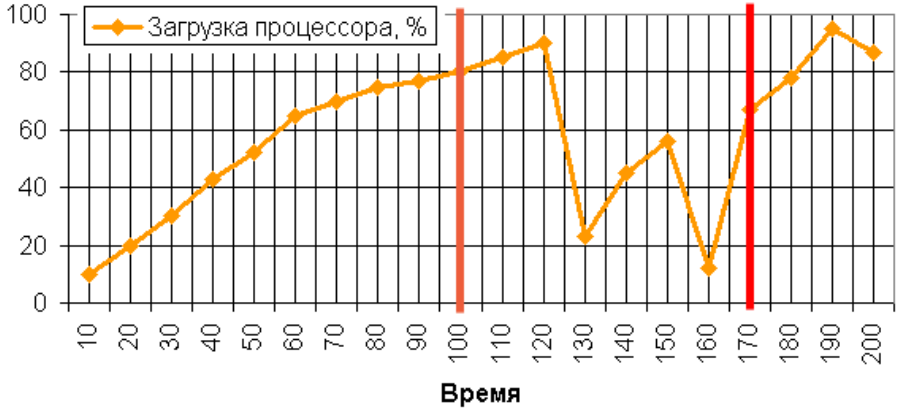
Примеры: rush up test

- Ужасный









Rush up test: задачи

- поиск и **определение параметров точки насыщения**;
- определение показателей масштабируемости системы;
- определение показателей нагрузки, при которых:
  - система работает заведомо стабильно;
  - система работает нестабильно;
  - система гарантированно выходит из строя;
- определение ключевых параметров аппаратной и сетевой конфигураций, влияющих на производительность.
- определение значений низкой, средней и высокой нагрузки для low-, mid-, high-load теста.

Пример: low-, mid-, high-load

- Приемлемый вариант при различном количестве пользователей:
  - **10-15%,**
  - **40-45%,**
  - **80-85%**

от количества пользователей, приводящих систему к точке насыщения.

- Выполнение тестирования с этими значениями на протяжении долгого времени, – «тестирование на выживаемость» (longevity test).

Low-, mid-, high-load тест: задачи

- Тестирование системы под различной нагрузкой позволяет определить её реакцию на соответствующие условия эксплуатации.
- Сбор статистической информации:
  - об использовании аппаратных и сетевых ресурсов и характере изменения этого использования;
  - о любых важных средних, минимальных, максимальных показателях и их дисперсии.

Longevity test : задачи

- Тесты на выживаемость системы показывают её способность **сохранять заданные показатели производительности, находясь длительное время под высокой нагрузкой .**

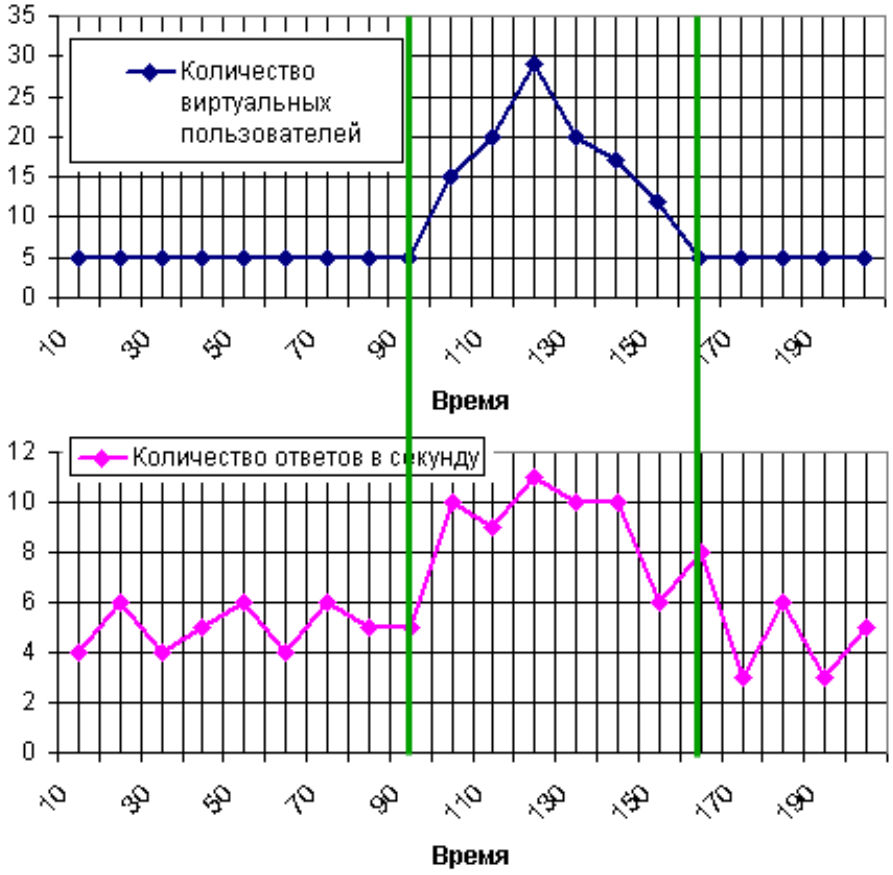
Примеры: Rush-hour test

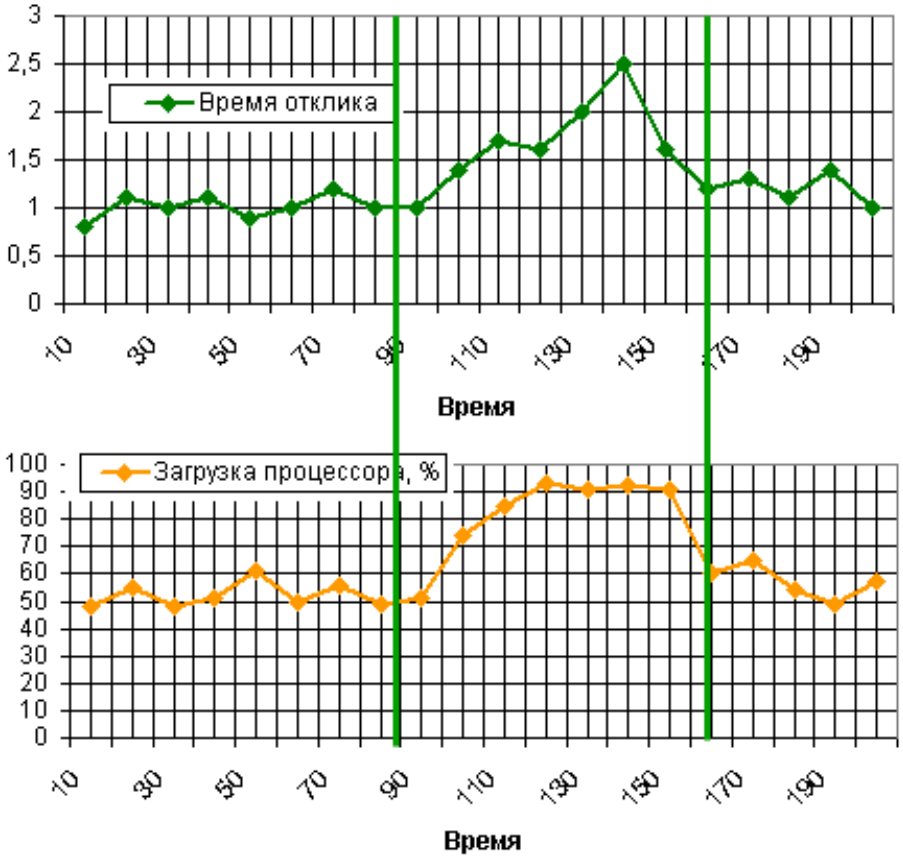
- Тест «часа пик» очень полезен в силу того, факта, что **он идеально отражает реальную жизнь**, когда в силу суточных колебаний интересов аудитории, календарных событий и др. нагрузка может очень резко изменяться.

Rush-hour test: задачи

- Основная задача rush-hour теста – проверить поведение системы **ВО ВРЕМЯ** всплеска нагрузки и **ПОСЛЕ НЕГО**.
- Полезен для исследования способности системы активировать свои «средства выживания под большой нагрузкой»

Примеры: Rush-hour test





Вопросы

1.Что такое «тестирование производительности»?

2.Каковы основные задачи ramp-up теста?

3.Каковы цели стрессового тестирования?

4.Что исследует объёмное тестирование?

5.Почему тестирование производительности важно?

6.Из каких характеристик складывается производительность?

7.Что исследует rush hour тест?

8.Какую информацию о системе позволяет собрать longevity тест?

9.Под какой нагрузкой, обычно, проводится longevity тест? Как определить эту нагрузку?

10.Почему для исследования производительности системы следует проводить много разнообразных тестов?

Этапы проведения тестирования производительности

Основные этапы

1. Сбор необходимой информации о системе;
2. Разработка модели нагрузки;
3. Выбор инструментальных средств;
4. Создание и отладка тестов;
5. Проведение тестирования;
6. Анализ результатов и формирование отчётности.

1. Сбор информации

Факторы, влияющие на производительность:

- работа с сетью;
- работа с БД;
- формирование интерфейса;
- кэширование;
- вычислительно сложные операции;
- обработка больших объёмов данных.

2. Разработка модели нагрузки

Модель нагрузки – совокупность сценариев работы с системой, успешное выполнение которых зависит от показателей производительности.

Необходимо определить:

- список тестируемых операций (ЧТО конкретно мы будем проверять);
- интенсивность выполнения операций (операции, выполняемые наиболее интенсивно, и операции, сильнее всего влияющие на производительность);
- зависимость изменения интенсивности выполнения операций от времени (параллельное выполнение операций; ситуации типа «час пик» и т.п.)

2. Разработка модели нагрузки

При выборе подвергаемых тестированию операций следует руководствоваться двумя показателями:

- снижение производительности этих операций уменьшает количество полезных действий пользователя в единицу времени (например, банк может обслужить меньше клиентов);
- снижение производительности этих операций ставит под угрозу работоспособность системы.

С технической точки зрения – это операции, интенсивно потребляющие аппаратные ресурсы.

2. Разработка модели нагрузки

- **Профиль нагрузки** – характерное для реальной жизни взаимодействие пользователей с системой, взаимозависимое с производительностью системы:
  - «утро, все пришли на работу и зашли на наш новостной сайт» (ситуация «час-пик»);
  - «стандартный день, ничего выдающегося»;
  - ???

2. Разработка модели нагрузки

Пересчёт показателей:

- *10 пользователей, выполняющих по 100 операций в секунду, создадут нагрузку, сопоставимую со 100 пользователями, выполняющими по 10 операций в секунду.*

3. Выбор инструментальных средств

Критерии:

- способность средства реализовывать необходимые профили и модели нагрузки;
- сложность формирования тестов в среде инструментального средства;
- форму отчётности, предоставляемую инструментальным средством;
- требования к аппаратной части «генератора нагрузки»;
- стоимость инструментального средства;
- возможность использования готовых решений или создания решений, которые можно будет использовать в последующих проектах.

3. Выбор инструментальных средств

Наиболее известные коммерческие инструментальные средства тестирования производительности:

- HP Performance Center (+ HP LoadRunner);
- Rational Performance Tester;
- SilkPerformer;
- TestComplete.

Наиболее известные некоммерческие:

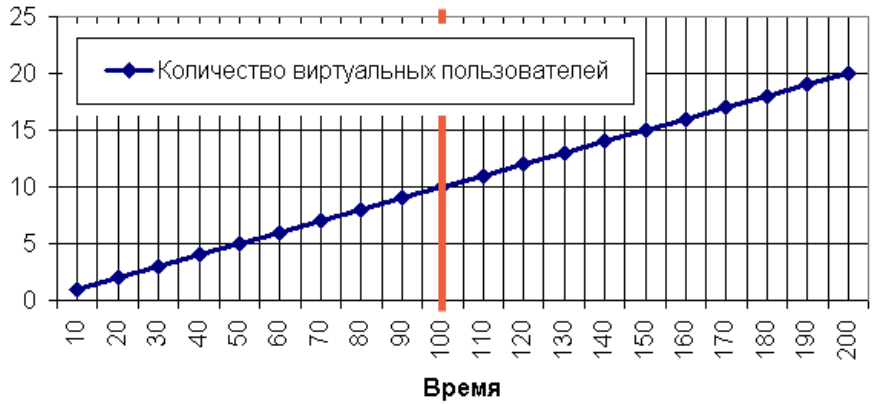
- Apache JMeter (!)
- Grinder.

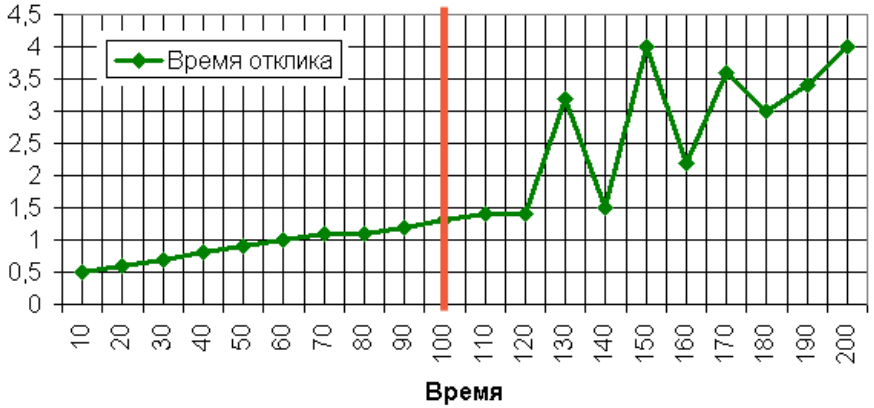
3. Проведение тестирования и отчётность

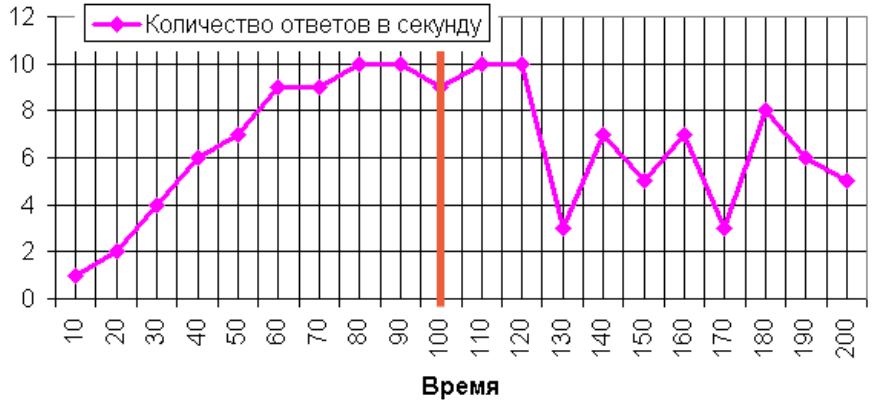
1. Тест: «*Запустим 250 виртуальных пользователей на три часа*». Что должен содержать отчет?
2. Тест: *определить точку насыщения*. Что должен содержать отчет? Параметры точки насыщения.
3. Rush-hour тест. Что должен содержать отчет? Описание поведения системы во всех ключевых моментах этого теста.
4. Longevity тест:
   - как долго система смогла выдерживать нагрузку;
   - как изменялась производительность системы, были ли утечки памяти, как быстро система «съедала» дисковое пространство и т.п.
5. ?

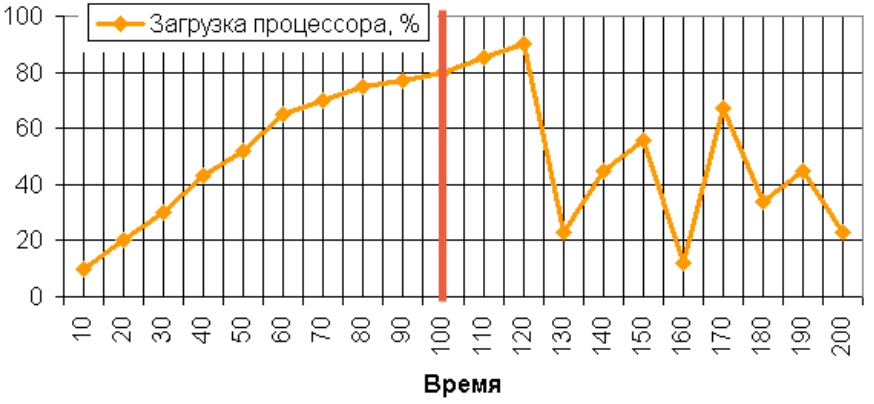
3. Проведение тестирования и отчётность

Ramp up тест.









Вопросы

1.Каковы основные этапы тестирования производительности?

2.Какую информацию о системе следует собрать до начала тестирования производительности?

3.Что такое модель нагрузки?

4.Что такое профиль нагрузки? Приведите примеры.

5.Что следует учитывать при выборе инструментальных средств тестирования производительности?

6.На что следует обращать особое внимание при подготовке отчёта о тестировании производительности?

7.Какая основная информация будет представлена в отчёте о rump up тесте?

8.Что такое (в контексте тестирования производительности) «узкое место»?

9.На какие функции системы следует обращать особое внимание при проведении тестирования производительности?

10.Существуют ли системы, для которых не имеет смысла проводить rush hour тест?