Подготовка к итоговому собеседованию по НТ

Вопросы:

1. Общие вопросы по НТ:
   1. Что такое и в чём заключается нагрузочное тестирование?
   2. На что стоит обратить внимание, если требуется выявить проблемы с производительностью системы?
   3. Каким образом, кроме метрик, можно определить появление проблем с производительностью? (корректность работы)?
2. Средства НТ:
   1. Какие инструменты по НТ ты знаешь?
   2. Если работал с LR, то с какими протоколами?
   3. Как сделать проверку по содержимому для WEB протокола?
   4. Как записать транзакцию в ошибочную?
   5. Что такое корреляция и параметризация в скриптах?
   6. Что такое Pacing? Зачем он нужен? Какие варианты настройки возможны?
   7. Что такое Think Time? Зачем он нужен?
   8. В чём отличие Pacing от Think Time?
   9. Распределённое тестирование (JMeter slave/master). Параллельная работа JMeter.
   10. JMETER:
       1. Что такое Assertion и зачем он нужен?
       2. Как устанавливать Pacing?
       3. Какие знаешь плагины?
3. Базы данных:
   1. Какие типы БД знаешь?
   2. О каких БД слышал/с какими работал?
4. Сети и протоколы:
   1. В чём отличие TCP от UDP?
   2. В чём отличие HTTP от HTTPs?
   3. На каком уровне в классической модели OSI работает протокол SSL?
   4. Из чего состоит HTTP-запрос?
5. Общие вопросы:
   1. Что такое веб-сервисы? Какие веб-серверы знаешь?
   2. Что такое SOAP/REST, wsdl?
   3. Работал ли с форматами XML и JSON?
6. Дополнительно:
   1. Регулярные выражения
   2. Как из строки из букв и из цифр вывести только цифры?
7. Что такое и в чём заключается нагрузочное тестирование?

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. В роли нагрузки может выступать количество пользователей (VU или кол-во Thread’ов), а также количество операций на сервере.

Тестирование производительности – тестирование, которое проводится с целью определения, как быстро работает вычислительная система или её часть под определённой нагрузкой. Также может служить для проверки и подтверждения других атрибутов качества системы, таких как масштабируемость, надёжность и потребление ресурсов.

Пиковая производительность – максимальная интенсивность работы системы.

Виртуальный пользователь – это последовательность действий, выполняемая скриптом, который эмулирует действия пользователя.

Надёжность системы – время, которое система может работать под высокой нагрузкой, а также ее способность к регенерации.

Тест завершается, когда

* 90percentil времён отклика вышел за SLA.
* количество неуспешных операций увеличилось до критического (более 5% за 10 минут);
* количество успешных операций не растет при увеличении нагрузки;
* исчерпаны системные или аппаратные ресурсы.

Тест подтверждения выполняется на уровне нагрузки:

* при тестировании целевых требований системы (первичное тестирование) - 80% от уровня найденной максимальной производительности.
* при тестировании релизов (повторное тестирование) - 110% от текущей пиковой производительности продуктива (основного профиля тестирования)
* при тестировании других задач использовать правило из п. 2

Длительность тестирования 1ч.

Критериями успешного завершения нагрузочного тестирования являются:

* Выполнение всех запланированных тестов;
* Получение данных мониторинга;

1. На что стоит обратить внимание, если требуется выявить проблемы с производительностью системы?

ОСНОВНЫЕ ПРОБЛЕМЫ ПРОИЗВОДИТЕЛЬНОСТИ

1. Сетевые ограничения;

2. Нехватка ресурсов аппаратной части (конфигурации аппаратной части);

3. Нехватка системных ресурсов для приложения;

4. Неэффективное использование ресурсов ПО (утечки памяти, параллелизм, механизмы использования потоков, старые технологии);

5. Архитектурные (при проектировании ПО и при проектирования БД)

6. Нелинейная масштабируемость системы;

7. Проблемы отказоустойчивости и надежности непрерывной эксплуатации (24/7);

8. Накопление ошибок и использования «костылей/заплаток» (рефакторинг кода и /или смена технологий);

9. Плавающие ошибки, ошибки производительности, возникающие только под нагрузкой

1. Каким образом, кроме метрик, можно определить появление проблем с производительностью? (корректность работы)?

Самостоятельно зайти на сайт и посмотреть на отклик, физически потрогать сервер на температуру.

Средства НТ:

1. Какие инструменты по НТ ты знаешь?

Для написания и запуска скриптов и тестов: Grinder, Gutling, Yandex.tank, LR (vugen, analysis, controller), Jmeter.

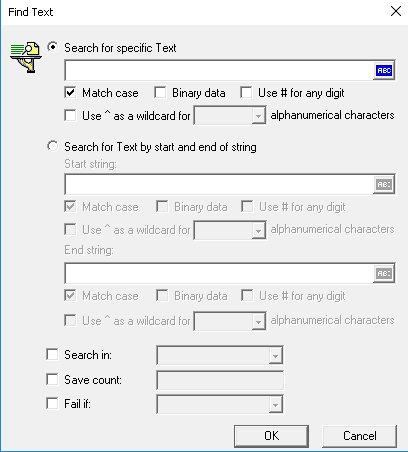
Для мониторинга: perfmon, telegraf, influxdb, Grafana.

1. Если работал с LR, то с какими протоколами?

WEB-HTTP/HTML

1. Как сделать проверку по содержимому для WEB протокола?

В step toolbox’е функция web\_reg\_find



1. Как записать транзакцию в ошибочную?

В web\_reg\_find в настройках fail if (FOUND/NOT FOUND)

Lr\_fail\_trans\_with\_error или LR\_FAIL

1. Что такое корреляция и параметризация в скриптах?

Корреляция – идентификаторы должны быть заменены на переменные, которые от запуска к запуску будут принимать корректные значения, извлекаемые из ответов на запросы к системе (например, идентификатор пользователя, полученный при входе в системе, который необходимо предъявлять во всех последующих транзакциях).

Параметризация – замена статичных или записанных данных на данные из подготовленного списка возможных вариантов), чтобы каждый виртуальный пользователь использовал надлежащий набор данных. «Надлежащий» означает, что данные разных виртуальных пользователей достаточно сильно различаются, чтобы избежать проблем с кешированием и конкурентным выполнением кода.

1. Что такое Pacing? Зачем он нужен? Какие варианты настройки возможны?

При планировании нагрузки важно выдержать заданную интенсивность работы теста. Это достигается за счёт того, что используется фиксированный шаг нагрузки — такой интервал времени, в течение которого гарантировано выполняется один проход сценария.

1) Запуск новой итерации сразу, как только завершится предыдущая итерация.

2) Запуск новой итерации с фиксированной или рандомной задержкой после того, как завершится предыдущая.

3) Запуск новой итерации через каждый фиксированный/рандомный интервал времени.

1. Что такое Think Time? Зачем он нужен?

Время ожидания между действиями. Позволяет эмулировать поведение реальных пользователей.

1. В чём отличие Pacing от Think Time?

Pacing – позволяет контролировать время между итерациями (БО), Think time позволяет контролировать время между действиями.

1. Распределённое тестирование (JMeter slave/master). Параллельная работа JMeter.

Запуск параллельных тестов с master на slave, а slave уже на целевой ресурс.

1. JMETER:
   * 1. Что такое Assertion и зачем он нужен?
     2. Как устанавливать Pacing?
     3. Какие знаешь плагины?

Assertion – инструмент, для поиска содержимого на странице ответа. Позволяет при нахождении и/или не нахождении какого-то содержимого на странице, позволяет записать транзакцию в ошибочную.

Для задания Pacing’а используются различные таймеры.

**jp@gc — Ultimate Thread Group, Constant Throughput Timer.**

**jp@gc – custom thread group, dummy sampler, listener pack (для связки инфлакса и графаны), perfmon, random csv dataset, jdbc.**

* + 1. Базы данных:

1. Какие типы БД знаешь?
2. О каких БД слышал/с какими работал?

Реляционные и нереляционные базы данных.

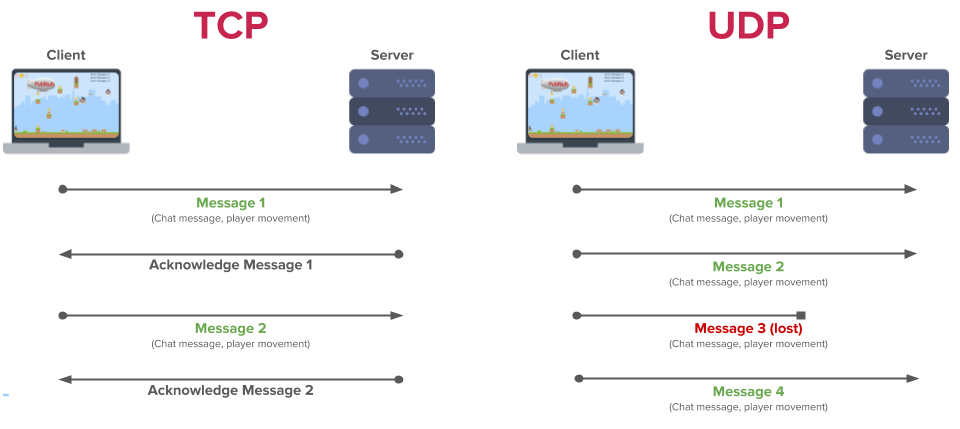
**Реляционная модель** - совокупность данных, состоящая из набора двумерных таблиц. В теории множеств таблице соответствует термин отношение (relation), физическим представлением которого является таблица, отсюда и название модели – реляционная.

**Нереляционная база данных** — это база данных, в которой в отличие от большинства традиционных систем баз данных не используется табличная схема строк и столбцов. В этих базах данных применяется модель хранения, оптимизированная под конкретные требования типа хранимых данных. Например, данные могут храниться как простые пары "ключ — значение", документы JSON, XML или граф, состоящий из ребер и вершин.  
influxdb, H2 (в jmeter), MySQL

Сети и протоколы:

* 1. В чём отличие TCP от UDP?

1. TCP гарантирует доставку пакетов данных в неизменном виде, последовательности и без потерь, UDP ничего не гарантирует.
2. TCP нумерует пакеты при передаче, а UDP нет
3. TCP работает в дуплексном режиме, в одном пакете можно отправлять информацию и подтверждать получение предыдущего пакета.
4. TCP требует заранее установленного соединения, UDP соединения не требует, у него это просто поток данных.
5. UDP обеспечивает более высокую скорость передачи данных.
6. TCP надежнее и осуществляет контроль над процессом обмена данными.
7. UDP предпочтительнее для программ, воспроизводящих потоковое видео, видеофонии и телефонии, сетевых игр.
8. UPD не содержит функций восстановления данных



* 1. В чём отличие HTTP от HTTPs?
* HTTPS не является отдельным протоколом передачи данных, а представляет собой расширение протокола HTTP с надстройкой шифрования;
* передаваемые по протоколу HTTP данные не защищены, HTTPS обеспечивает конфиденциальность информации путем ее шифрования;
* HTTP использует порт 80, HTTPS — порт 443.

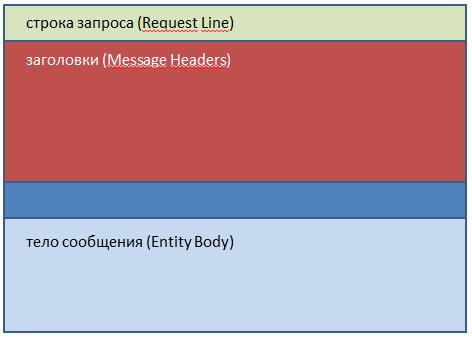
https – надстройка над протоколом HTTP, использующая шифрование TLS или SSL.

* 1. На каком уровне в классической модели OSI работает протокол SSL?

Между прикладным и транспортным.

Протокол **SSL** имеет два **уровня**: протокол записей (**SSL** Record Protocol) и протокол диалога (**SSL** Handshake Protocol).

* 1. Из чего состоит HTTP-запрос?



**Строка запроса** – указывает метод передачи, URL-адрес, к которому нужно обратиться и версию протокола HTTP.

**Заголовки** – описывают тело сообщений, передают различные параметры и др. сведения и информацию.

**Тело сообщения** - это сами данные, которые передаются в запросе.  Тело сообщения – это необязательный параметр и может отсутствовать.

Общие вопросы:

* 1. Что такое веб-сервисы? Какие веб-серверы знаешь?

Веб-сервис – это технология для обмена информацией между программами, могут быть написаны на разных языках и могут быть расположены на разных узлах сети. Общение как надстройка протокола HTTP. Веб-сервис – адрес/ссылка, обращение к которой позволяет выполнить определенные действия и получить какой-то ответ.

Apache – для динамически содержимого, nginx – для статического.

* 1. Что такое SOAP/REST, wsdl?

SOAP (Simple object access protocol) – при использовании данные передаются в формате XML. Стандартизация, строгая сертификация, однозначность.

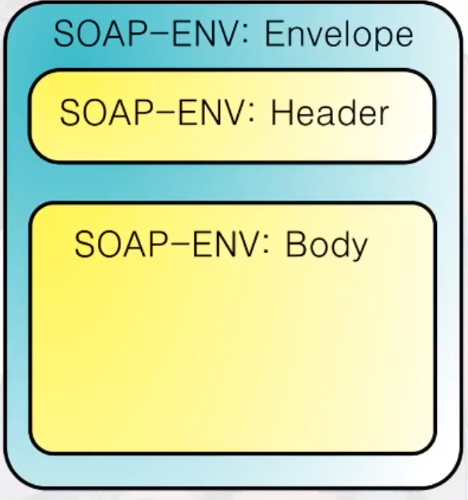
Состав SOAP:

-Envelope – корневой элемент – определение конца и начала сообщения.

-Header – необязательный – обработка основного сообщения.

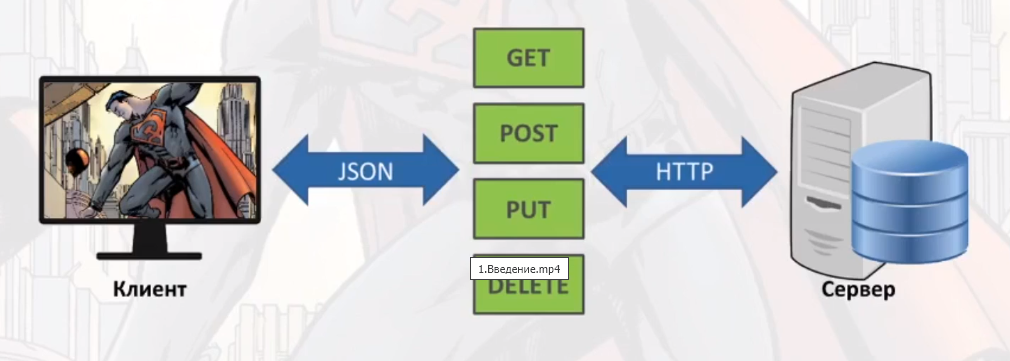
-Body – обязательный – вся основная информация сообщения.

-Fault – необязательный – для ошибок



REST (Representation state transfer) – нет официального стандарта – не использует конвертацию данных при передачи. Для управления данными используются HTTP-методы:

Get, post, put, delete



**WSDL** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) ***W****eb****S****ervices****D****escription****L****anguage* [/ˈwɪz dəl/](https://ru.wikipedia.org/wiki/%D0%A2%D0%B0%D0%B1%D0%BB%D0%B8%D1%86%D0%B0_%D0%9C%D0%A4%D0%90_%D0%B4%D0%BB%D1%8F_%D0%B0%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%BE%D0%B3%D0%BE_%D1%8F%D0%B7%D1%8B%D0%BA%D0%B0)) — язык описания [веб-сервисов](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81) и доступа к ним, основанный на языке [XML](https://ru.wikipedia.org/wiki/XML).

**<message** name="getTermRequest"**>**

**<part** name="term" type="xs:string"**/>**

**</message>**

**<message** name="getTermResponse"**>**

**<part** name="value" type="xs:string"**/>**

**</message>**

**<portType** name="glossaryTerms"**>**

**<operation** name="getTerm"**>**

**<input** message="getTermRequest"**/>**

**<output** message="getTermResponse"**/>**

**</operation>**

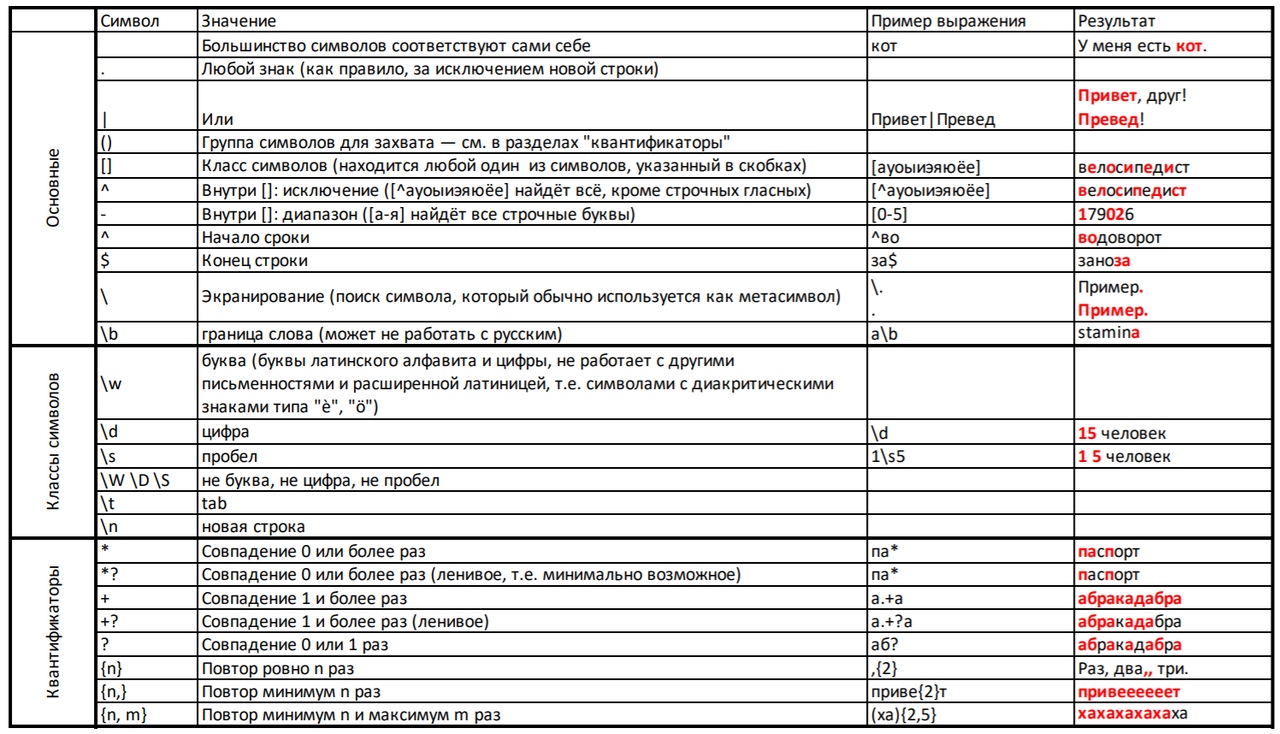
**</portType>**

* 1. Работал ли с форматами XML и JSON?

Да, отправлял SOAP запрос в скрипте в JMeter.

Да, аналогично использовал в скриптах, в fiddler’е отправлял запросы при написании заглушек и проверки ответа.

1. Регулярные выражения



1. Как из строки из букв и из цифр вывести только цифры?

\d+

SQL структура

SELECT ('столбцы или \* для выбора всех столбцов; обязательно')

FROM ('таблица; обязательно')

WHERE ('условие/фильтрация, например, city = 'Moscow'; необязательно')

GROUP BY ('столбец, по которому хотим сгруппировать данные; необязательно') HAVING ('условие/фильтрация на уровне сгруппированных данных; необязательно') ORDER BY ('столбец, по которому хотим отсортировать вывод; необязательно')

Обратная сортировка DESC

JOIN структура(inner/left/right/full)

select \*

from Orders JOIN Customers ON Orders.CustomerID = Customers.CustomerID