

# Как формулировать требования для нагрузочного тестирования?

Алексей Рагозин  
[alexey.ragozin@gmail.com](mailto:alexey.ragozin@gmail.com)

# О докладчике



Занимаюсь высоконагруженными системами на Java с 2006. Разрабатывал софт для торговли на фондовых рынках, телекоме, e-commerce, RTB, здравоохранения. Выступаю на конференциях, организую митапы, провожу тренинги.

Email: [alexey.ragozin@gmail.com](mailto:alexey.ragozin@gmail.com)

Blog: [blog.ragozin.info](http://blog.ragozin.info)

Github: <https://github.com/aragozin>

Митапы и вебинары: <https://aragozin.timepad.ru>

# О чём этот доклад?

Многие IT системы относятся к классу  
систем массового обслуживания (СМО)

*Микросервисы, веб приложения и пр.*

СМО довольно давно изучаются математикой

*“Queueing theory”, A. Erlang – 1909, “Исследование операций” – начало XX века*

И в теории, теория не должна отличаться от практики,  
но на практике это не всегда так.

# Зачем это знать аналитику?

## Нефункциональные требования (НФТ)

- Рамочные требования к производительности системы

*Нефункциональные требования прямо или косвенно обусловлены бизнес задачами системы*

- Проработка профилей нагрузки

*Реализация тестов требует информации о функционировании системы: ожидаемое поведение пользователей, цепочки вызовов микросервисов и т.д.*

# Виды тестирования

## Тестирование производительности

- Load testing – *нагрузочное тестирование*
- Volume testing – *объёмное тестирование*
- Stress testing – *стрессовое тестирование*
- Endurance testing – *тестирование стабильности*
- Scalability testing – *тестирование масштабируемости*

# Виды тестирования

## Тестирование производительности

- Load testing – *нагрузочное тестирование*
- Volume testing – *объёмное тестирование*
- Stress testing – *стрессовое тестирование*
- Endurance testing – *тестирование стабильности*
- Scalability testing – *тестирование масштабируемости*

# Системы массового обслуживания

(определение из учебника <http://mathhelpplanet.com/static.php?p=sistema-massovogo-obsluzhivaniya> )

Математическая модель системы массового обслуживания (СМО) включает три основных элемента:

- *поток поступающих сообщений,*
- *систему обслуживания,*
- *характеристики качества и дисциплину обслуживания.*

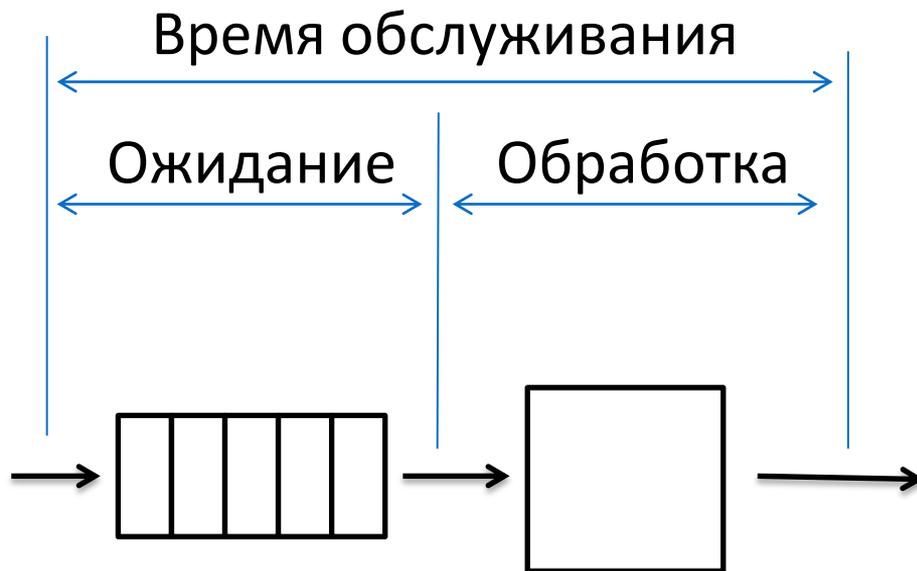
# Системы массового обслуживания

(определение из учебника <http://mathhelpplanet.com/static.php?p=sistema-massovogo-obsluzhivaniya> )

Математическая модель системы массового обслуживания (СМО) включает три основных элемента:

- **поток поступающих сообщений,**  
*интенсивность, статистическое распределение*
- **систему обслуживания,**  
*время обработки запроса*
- **характеристики качества и дисциплину обслуживания.**  
*FIFO/LIFO/..., таймауты*

# СМО с очередью



# Поток случайных событий

Равномерный поток событий

Нагрузка: **100 RPS**



# Поток случайных событий

Равномерный поток событий

Нагрузка: **100 RPS**



Пуассоновский поток - *поток независимых событий*



# Поток случайных событий

Равномерный поток событий

Нагрузка: **100 RPS**



Пуассоновский поток - *поток независимых событий*



# Поток случайных событий

Равномерный поток событий

Нагрузка: **100 RPS**



Пуассоновский поток - *поток независимых событий*



# Поток случайных событий

Равномерный поток событий

Нагрузка: **100 RPS**



Пуассоновский поток - *поток независимых событий*



# Поток случайных событий

Равномерный поток событий

Нагрузка: **100 RPS**



Пуассоновский поток - *поток независимых событий*



# Поток случайных событий

Равномерный поток событий

Нагрузка: **100 RPS**



Пуассоновский поток - *поток независимых событий*



# Поток случайных событий

Равномерный поток событий

Нагрузка: **100 RPS**



Пуассоновский поток - *поток независимых событий*



# Поток случайных событий

Равномерный поток событий

Нагрузка: **100 RPS**

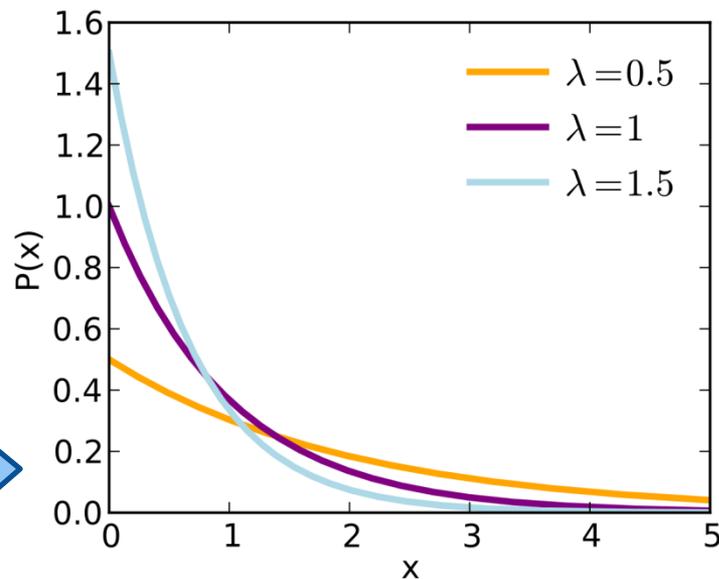
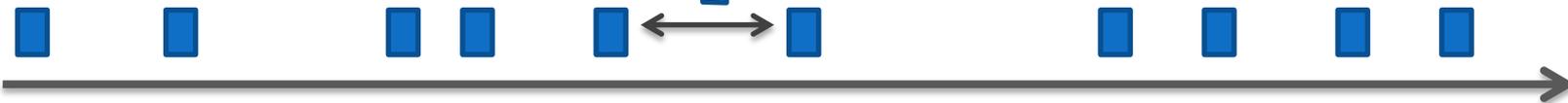


Пуассоновский поток - *поток независимых событий*

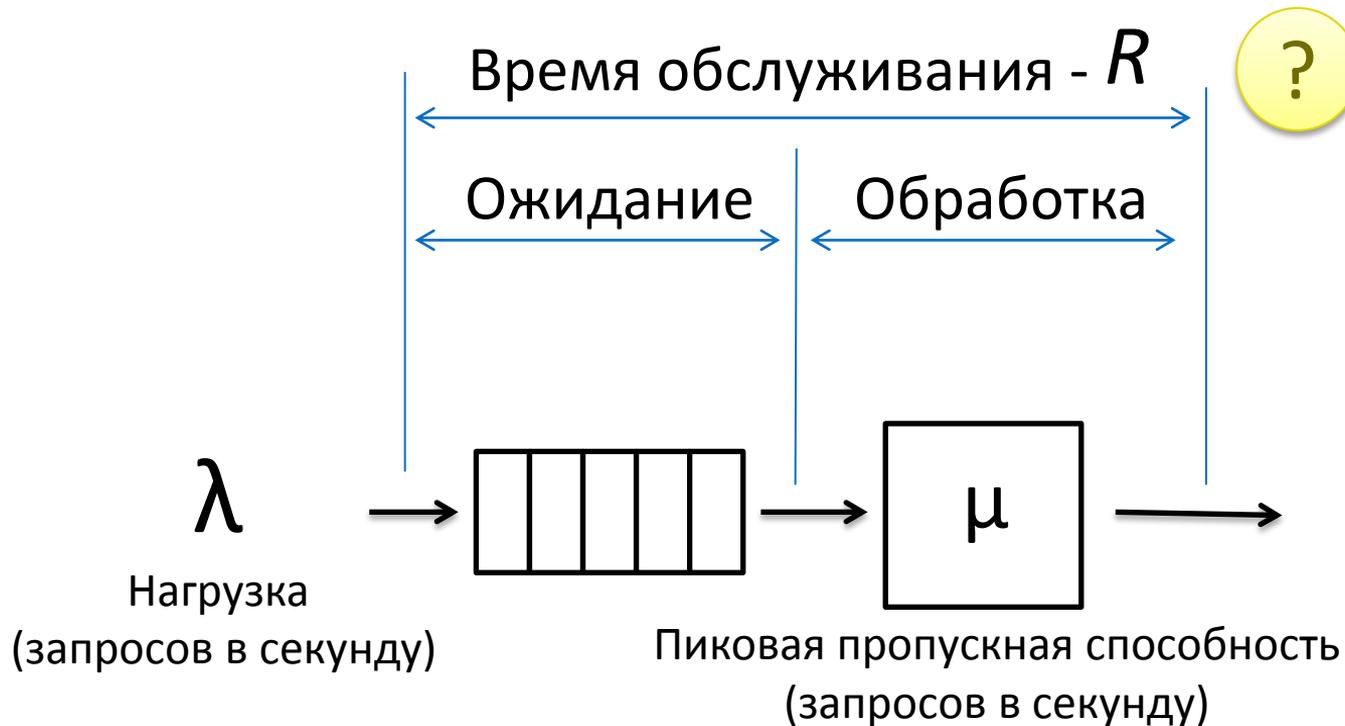


# Поток случайны событий

Пуассоновский поток



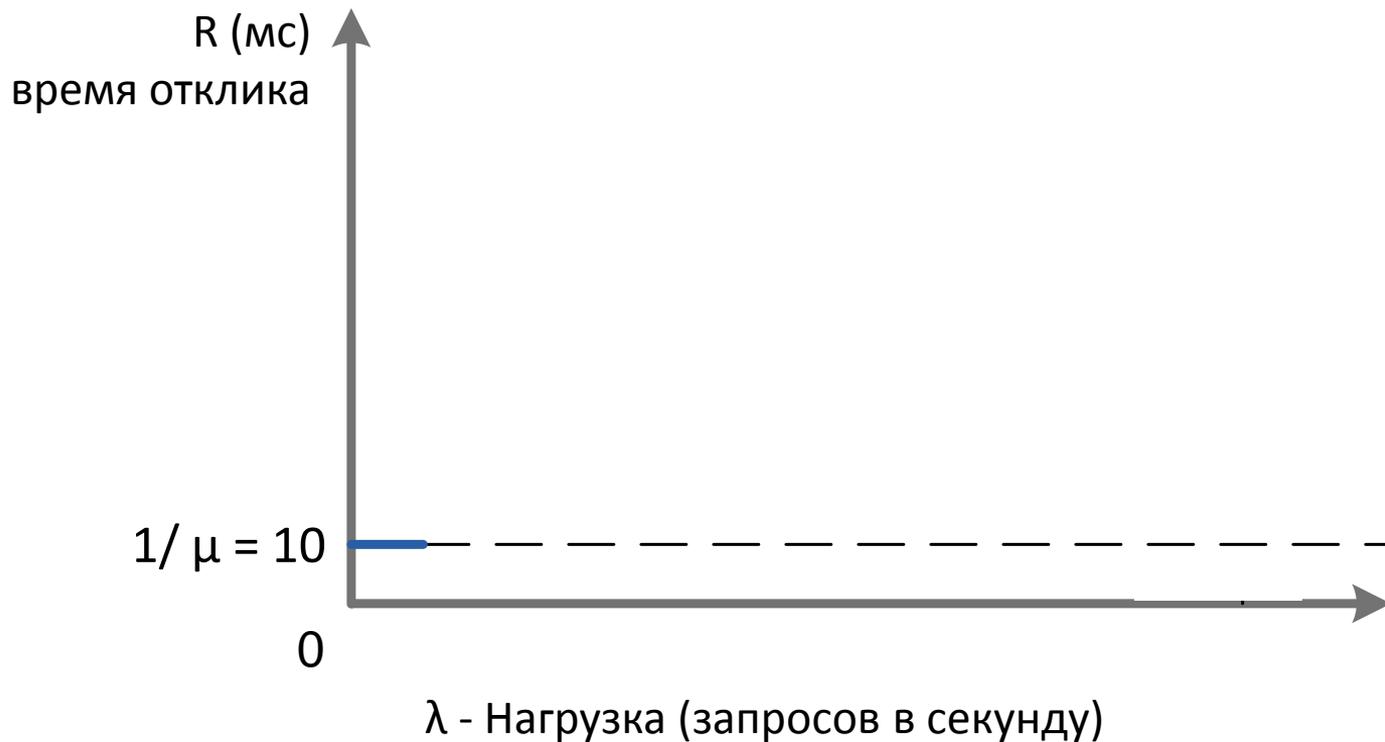
# СМО с очередью



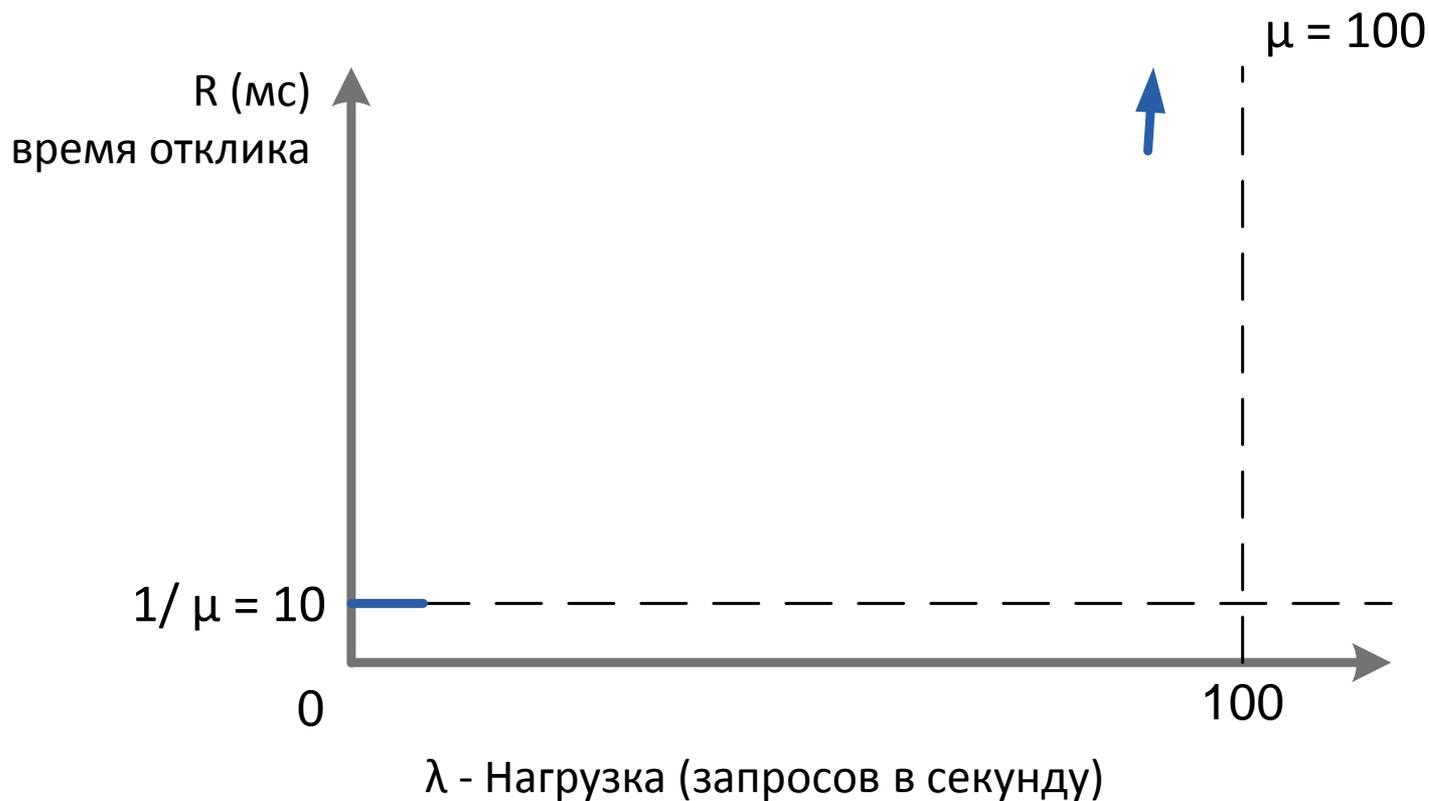
# СМО с очередью



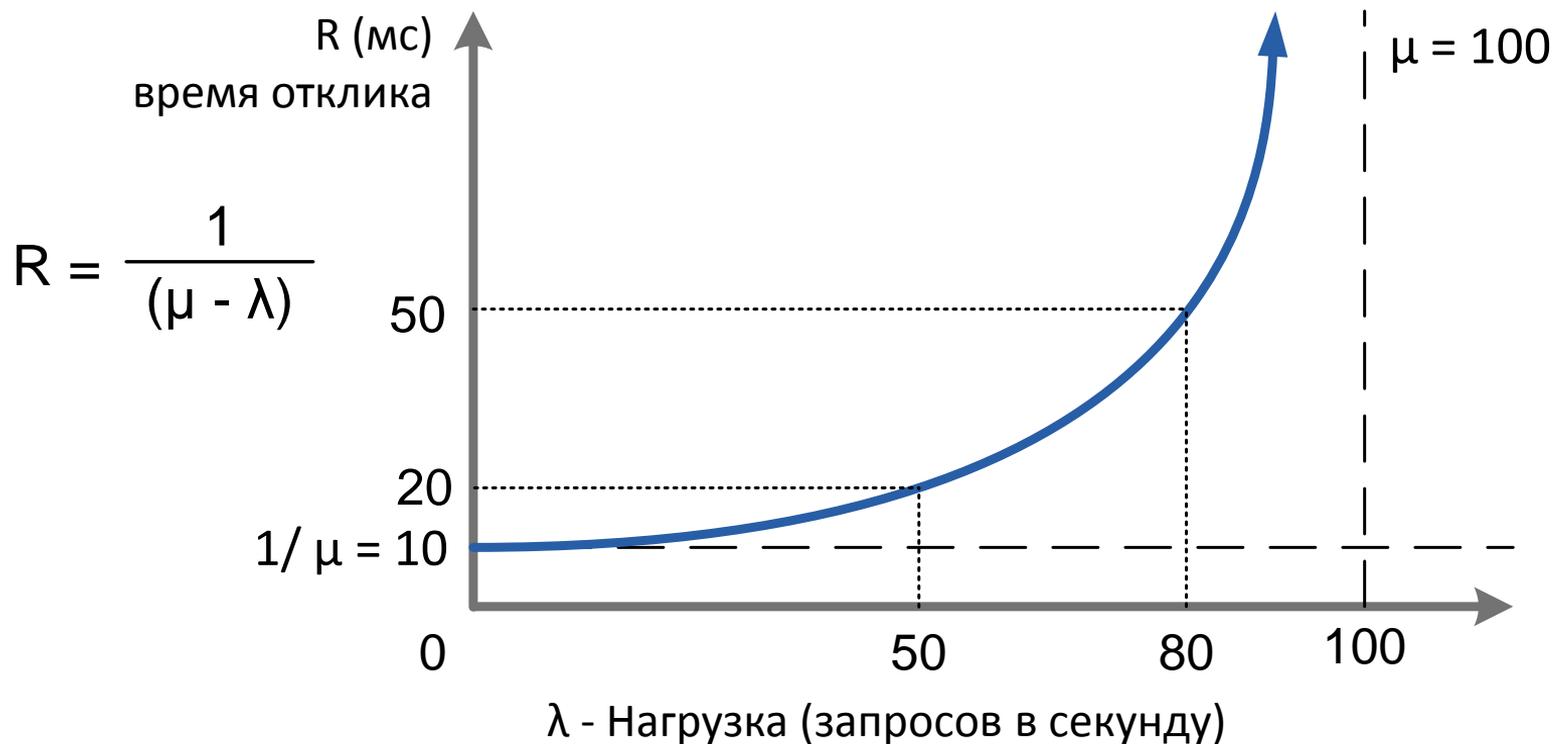
# СМО с очередью



# СМО с очередью

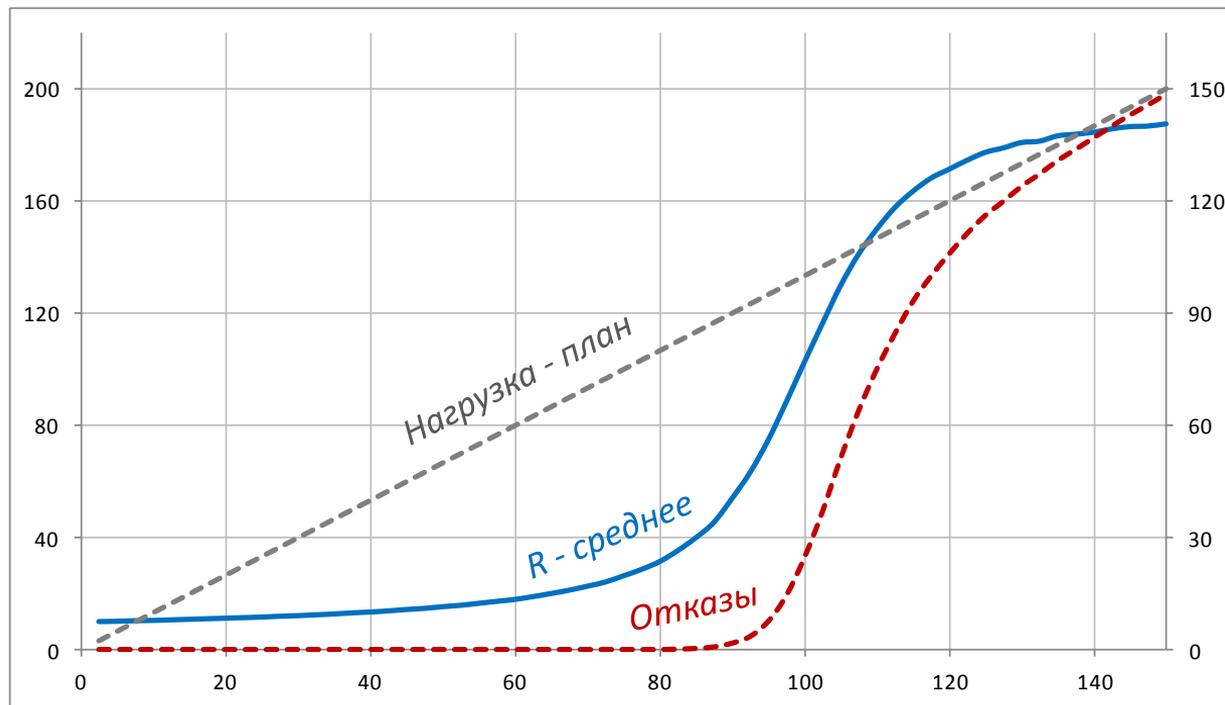


# СМО с очередью



# СМО с очередью

$R$  [мс]



$\lambda$  [1/с]

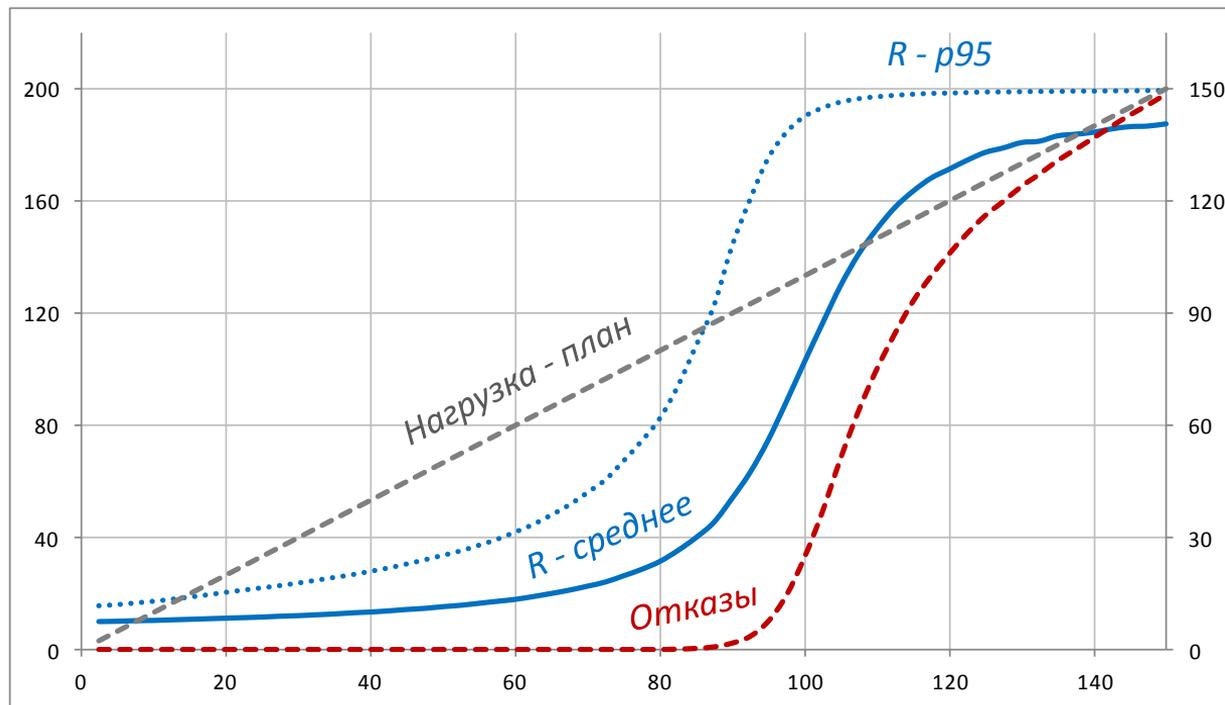
$\mu = 100$

$T_0 = 200$ мс

Нагрузка [1/с]

# СМО с очередью

$R$  [мс]



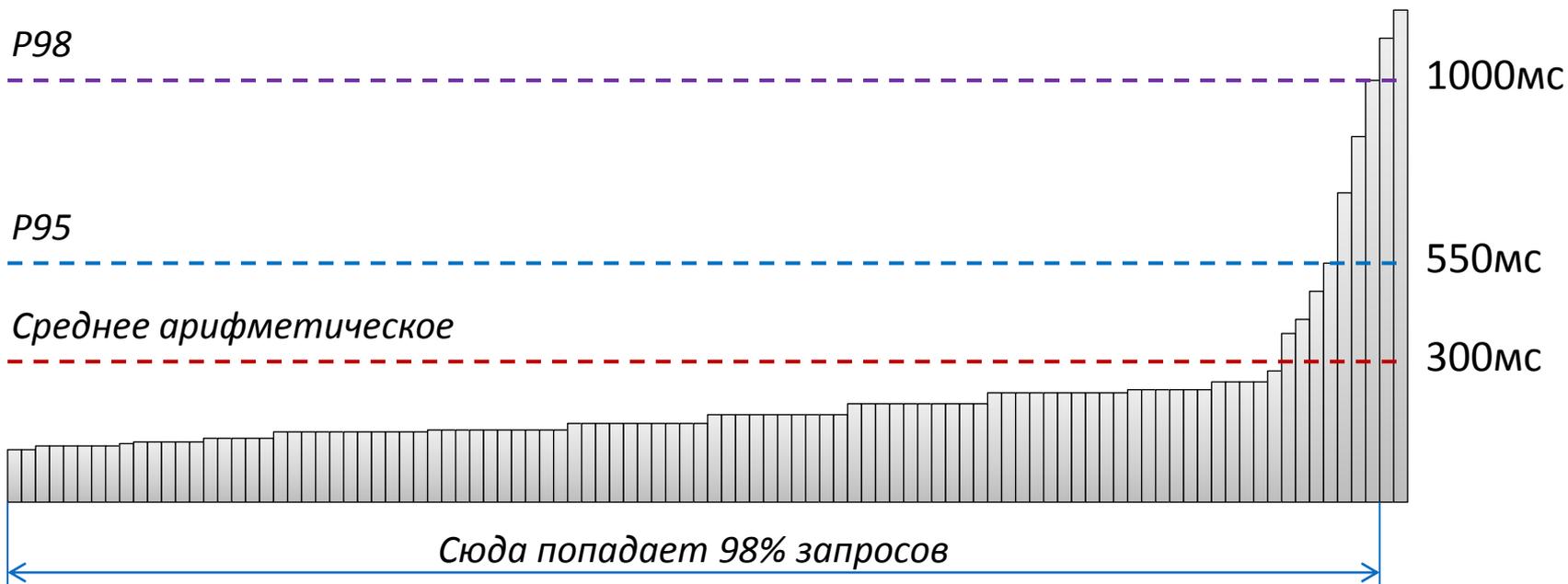
$\lambda$  [1/c]

$\mu = 100$

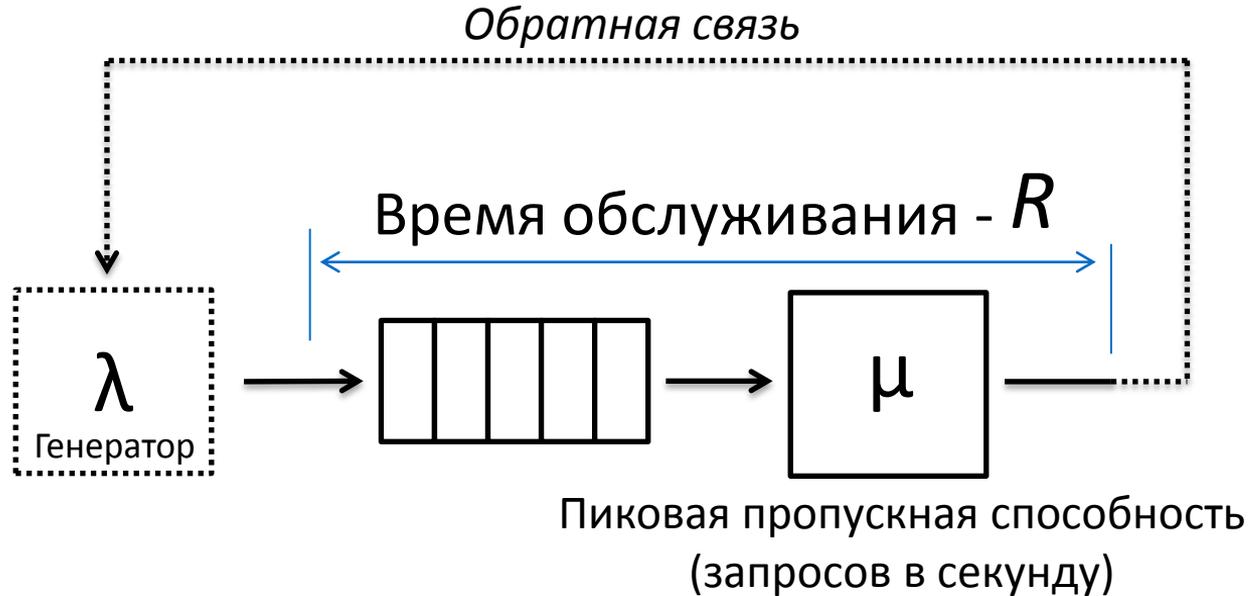
$TO = 200\text{мс}$

Нагрузка [1/c]

# Почему перцентили?

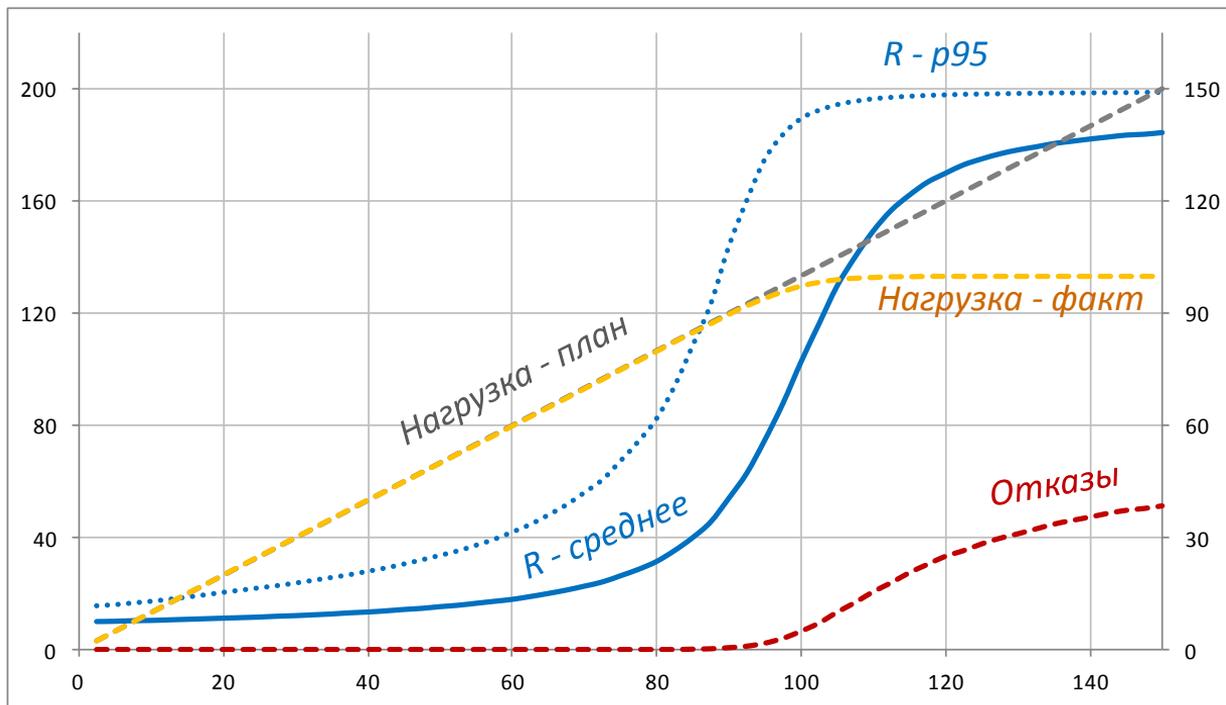


# Закрытая модель нагрузки



# Закрытая модель нагрузки

$R$  [мс]



$\lambda$  [1/с]

$\mu = 100$

$T_0 = 200\text{мс}$

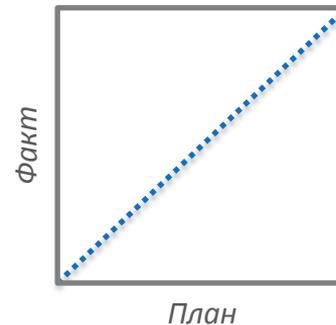
**Нагрузка  
20 - потоков**

Нагрузка [1/с]

# Модели нагрузки

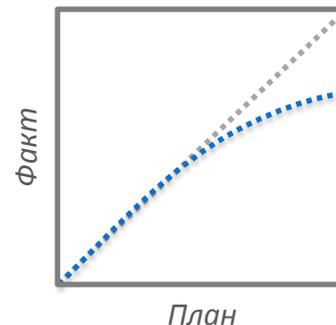
## Открытая модель нагрузки

- Поток запросов не зависит от реакции системы



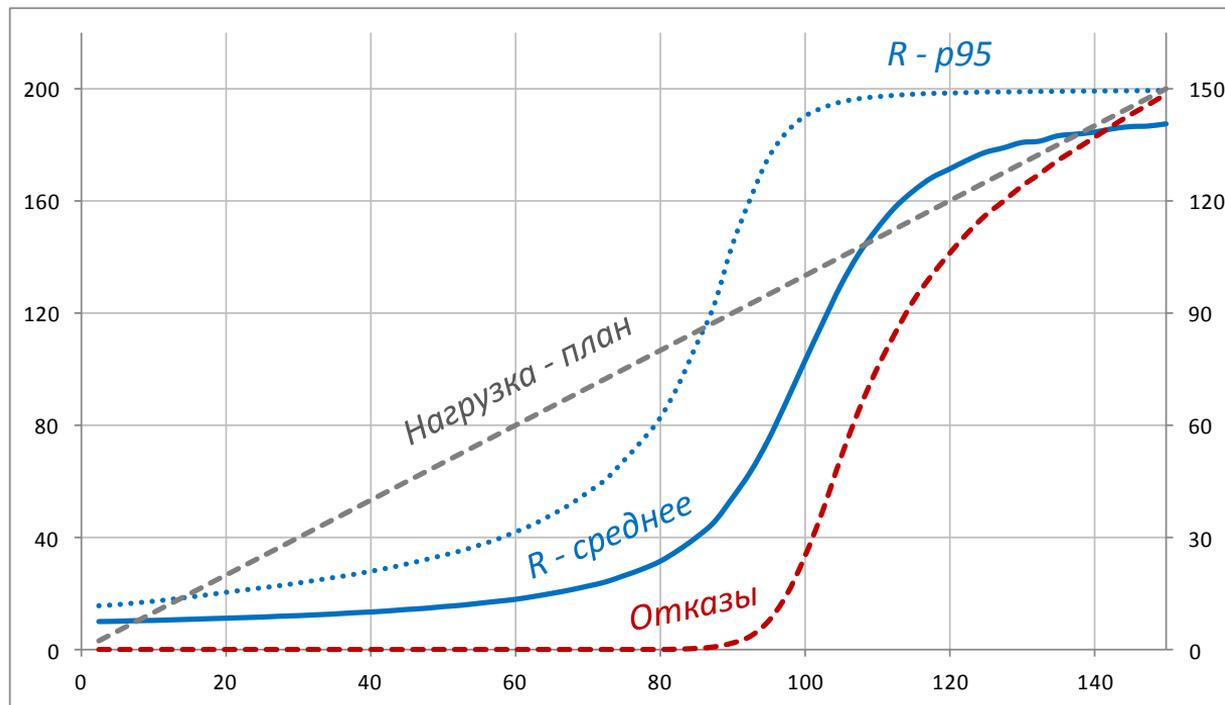
## Закрытая модель нагрузки

- Поток запросов реагирует на поведение системы



# СМО с очередью

$R$  [мс]



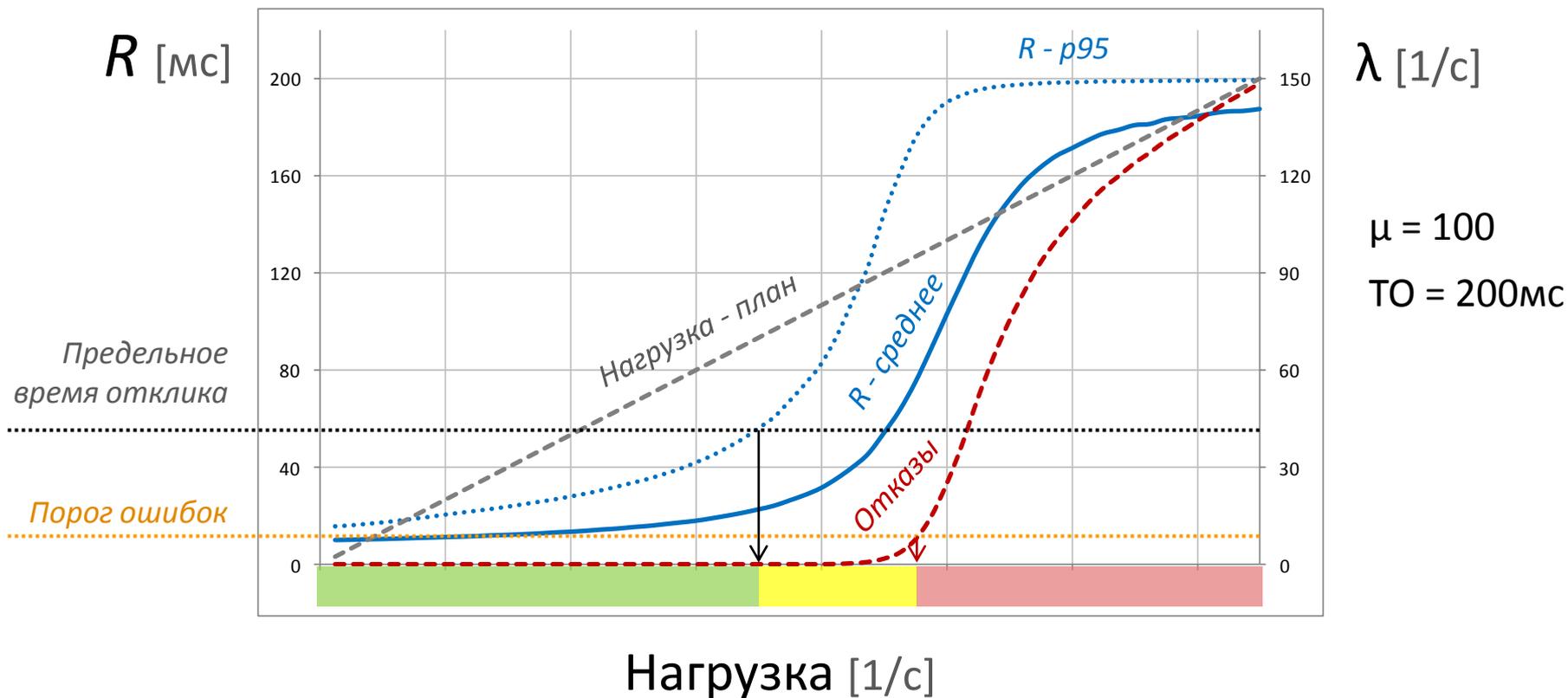
$\lambda$  [1/с]

$\mu = 100$

$T_0 = 200\text{мс}$

Нагрузка [1/с]

# СМО с очередью

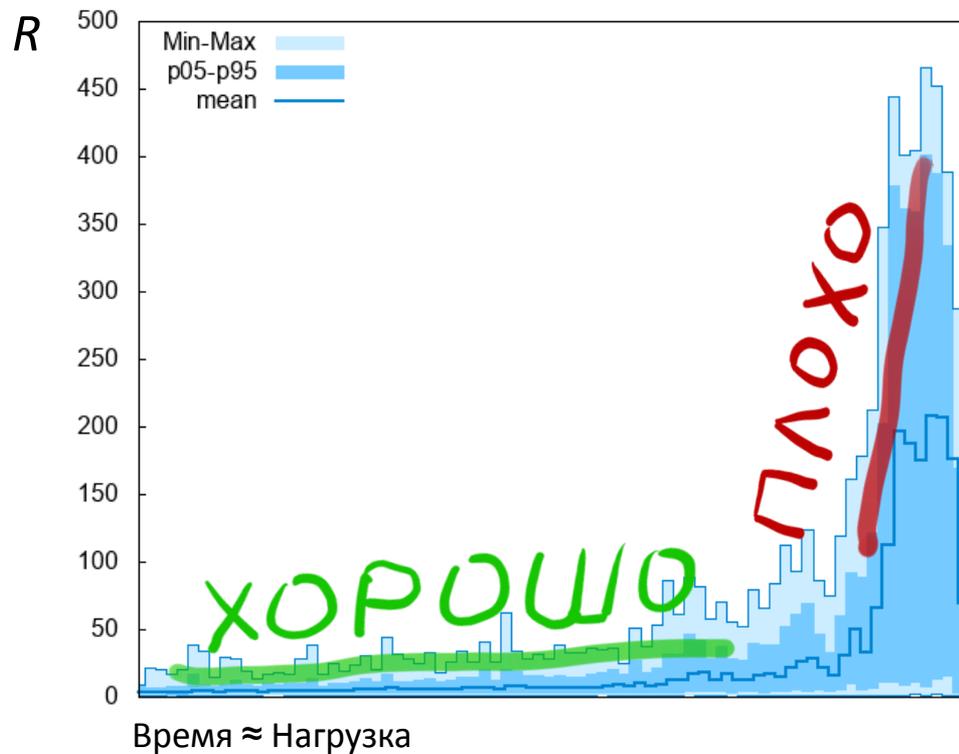


# Тестирование с линейной нагрузкой

## Тестирование с линейновозрастающей нагрузкой

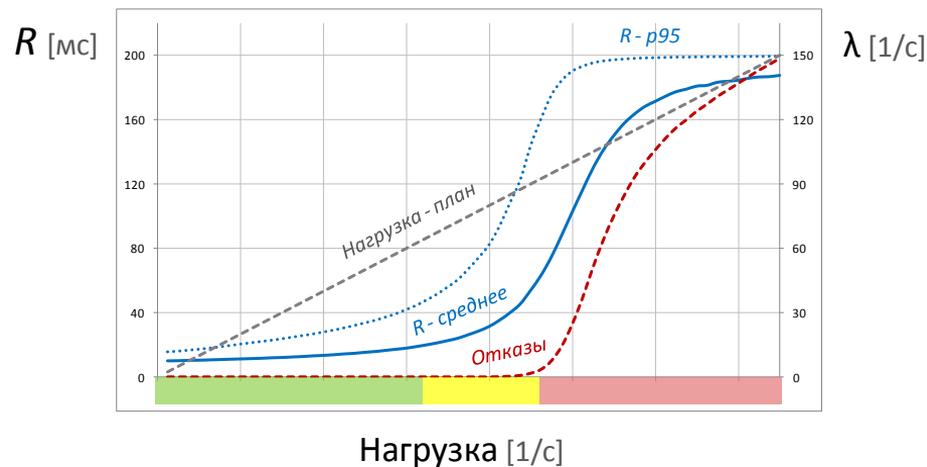
- Нагрузка линейно возрастает со временем
- Нарушение SLA по времени отклика → **точка деградации**
- Нарушение SLA по **ошибкам** → **предельно допустимая нагрузка**

# Точка деградации

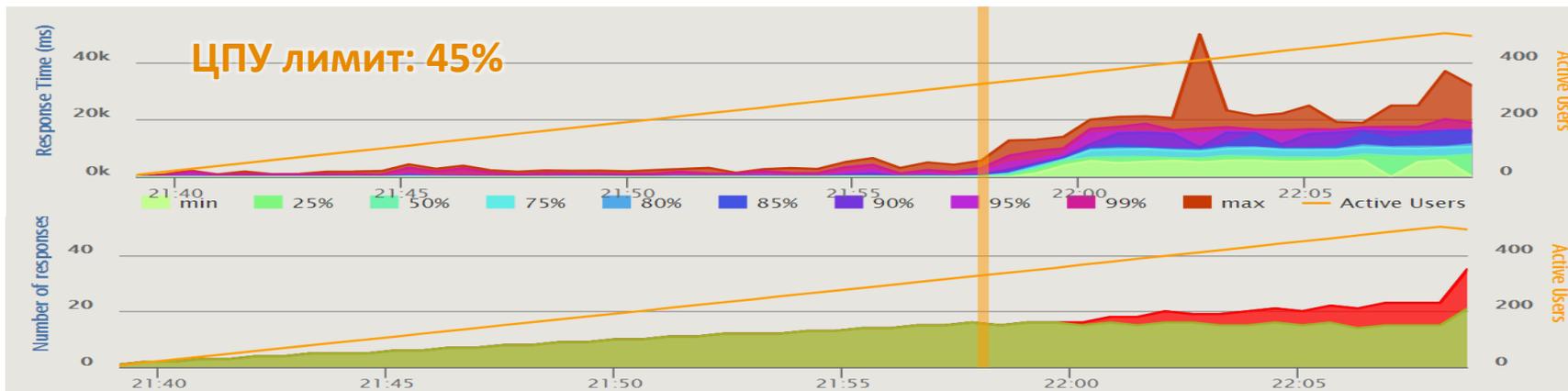
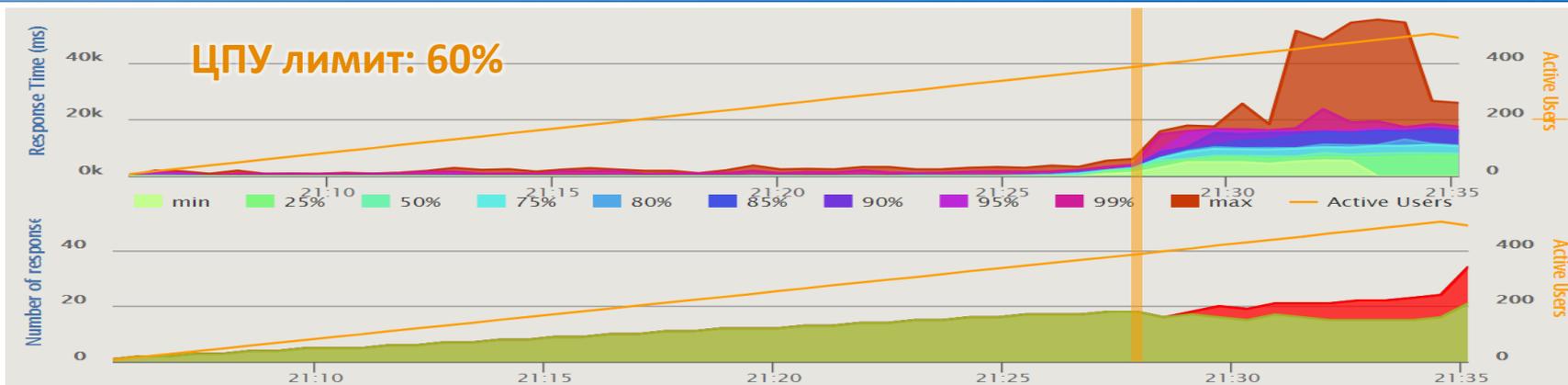


# СМО с очередью - ВЫВОДЫ

- СМО **не может работать на 100%** пропускной способности
- В **зелёной** зоне качество обслуживания определяется логикой обработки запроса
- В **жёлтой** зоне качество обслуживания определяется поведение очереди
- Ваша задача определить рабочую точку системы, задача разработки – чтобы она была в **зелёной** зоне



# Пропускная способность



Переходим к практике

# Как формулировать НФТ?

## Прямые требования

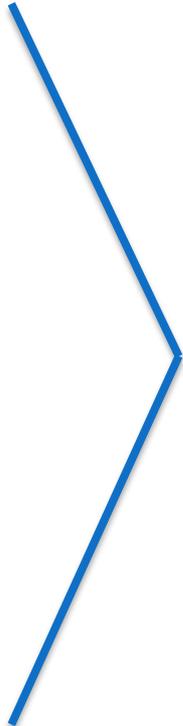
- Внешние SLA
- Внутренние SLA

## Косвенные требования

- Например: таймаут браузера

## Транзакционные объёмы

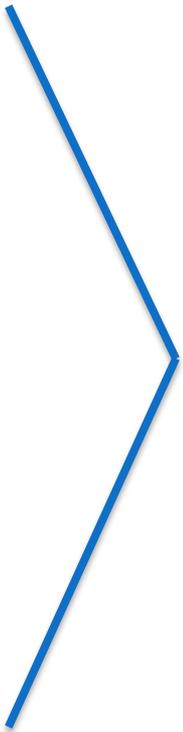
- Например: дневной объём операций



**НФТ**

# Как формулировать НФТ?

НФТ



## Интенсивность нагрузки

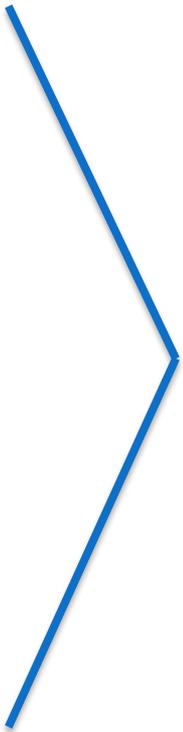
- Интенсивность
- Профиль нагрузки

## Качество обслуживания

- Время отклика (Например: **P99 < 200мс**)
- Допустимый уровень ошибок

# Как формулировать НФТ?

НФТ



Рабочая точка \*

## Интенсивность нагрузки

- Интенсивность
- Профиль нагрузки

## Качество обслуживания

- Время отклика (Например: **P99 < 200мс**)
- Допустимый уровень ошибок

\* Рабочих точек может быть несколько

# Интенсивность и качество

## Интенсивность нагрузки

- **Количественная** характеристика
- Должна задаваться скалярной величиной

## Качество обслуживания

- **Качественная** характеристика
- *“Не бывает осетрины второй свежести”*

# Гомогенная нагрузка

## Гомогенная нагрузка

*Поток однотипных несвязанных запросов*

- Интенсивность потока (RPS)



# Гетерогенная нагрузка

## Гетерогенная нагрузка

*Комбинация нескольких однородных потоков*

- Профиль нагрузки (70% чтение, 30% запись)
- Интенсивность потока (RPS)



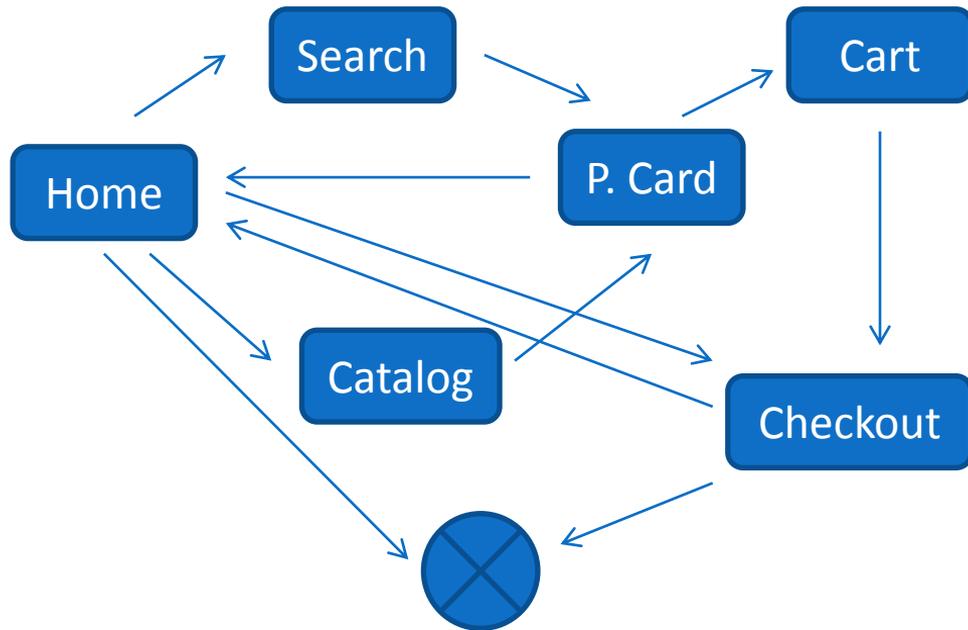
# Сценарная нагрузка

## Сценарная нагрузка

*Запросы взаимосвязаны сценарием поведения пользователя*



# Пользователь в магазине



access.log

```
... index.html
... search.html
... product.html
... cart/add
... index.html
... catalog.html
... checkout.html
... index.html
```

# Виртуальный пользователь

## Виртуальный пользователь (ВИ / VU)

### *Скрипт взаимодействия с системой*

- Взаимодействует с системой с заданной интенсивностью
- Имеет внутреннюю логику и состояние  
Например в форме виде марковского процесса
- Может использоваться реальный браузер

# Сценарная нагрузка

## Сценарная нагрузка

*Запросы взаимосвязаны сценарием поведения пользователя*

- Профиль нагрузки – скрипт поведения и его параметры
- Интенсивность – число одновременно обслуживаемых пользователей

# Другие типы нагрузки

## Асинхронная

- Несколько точек взаимодействия с тестируемой системой

## Пакетная

- Важна пропускная способность

# Типы тестирования

## Приёмочное тестирование

- Тестовый стенд в масштабе 1:1
- Тестируется каждая **рабочая точка**
- Проверяется SLA (прошёл / не прошёл)

*Цель – проверить на соответствие НФТ*

# Типы тестирования

## Тестирования на предельную нагрузку

- Результат - величина предельно допустимой нагрузки
- Методика - тестирования с линейновозрастающей или ступенчатой нагрузкой

*Цель - оценить уровень нагрузки  
в точке деградации  
при фиксированных ресурсах*



# Типы тестирования

## Регрессионное тестирование

- Выполняется на **стенде уменьшенного масштаба** (как правило)
- Измеряется уровень нагрузки в **точке деградации** для стенда
- Результат сравнивается с базой (результатом одной из предыдущих версий на **том же стенде**)

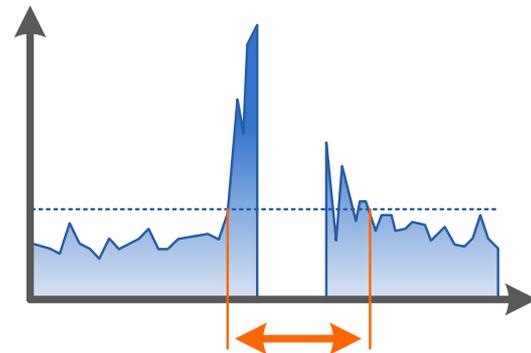
*Цель – оценить риски регрессии производительности в среде эксплуатации*

# Типы тестирования

## Стресс тестирование

- На стенд временно подаётся запроектная нагрузка
- Оценивается время восстановления (нормализация качества обслуживания)

*Цель – оценить  
время восстановления после сбоя*



# Выводы

Есть фундаментальные законы определяющие поведение информационных системы под нагрузкой!

# Выводы

Есть фундаментальные законы определяющие поведение информационных системы под нагрузкой!

Тестирование не возможно без требований.

- Корректно сформулированных
- Учитывающих все аспекты работы системы

# Выводы

Есть фундаментальные законы определяющие поведение информационных системы под нагрузкой!

Тестирование не возможно без требований.

- Корректно сформулированных
- Учитывающих все аспекты работы системы

При сценарной нагрузки, профиль – ключевой элемент.

# Выводы

Есть фундаментальные законы определяющие поведение информационных системы под нагрузкой!

Тестирование не возможно без требований.

- Корректно сформулированных
- Учитывающих все аспекты работы системы

При сценарной нагрузки, профиль – ключевой элемент.

Полноценное тестирования производительности – это сложно,  
начинайте двигаться малыми шагами.



# Спасибо!

## Алексей Рагозин

Email: [alexey.ragozin@gmail.com](mailto:alexey.ragozin@gmail.com)

Github: <https://github.com/aragozin>

Митапы: <https://aragozin.timepad.ru>

Учебный курс: <https://aragozin.timepad.ru/event/3042811/>