

# Распределённые трассировки

Какую (ещё) пользу можно извлечь  
из распределённых трассировок?

Григорий Кошелев

Контур

# Немного обо мне

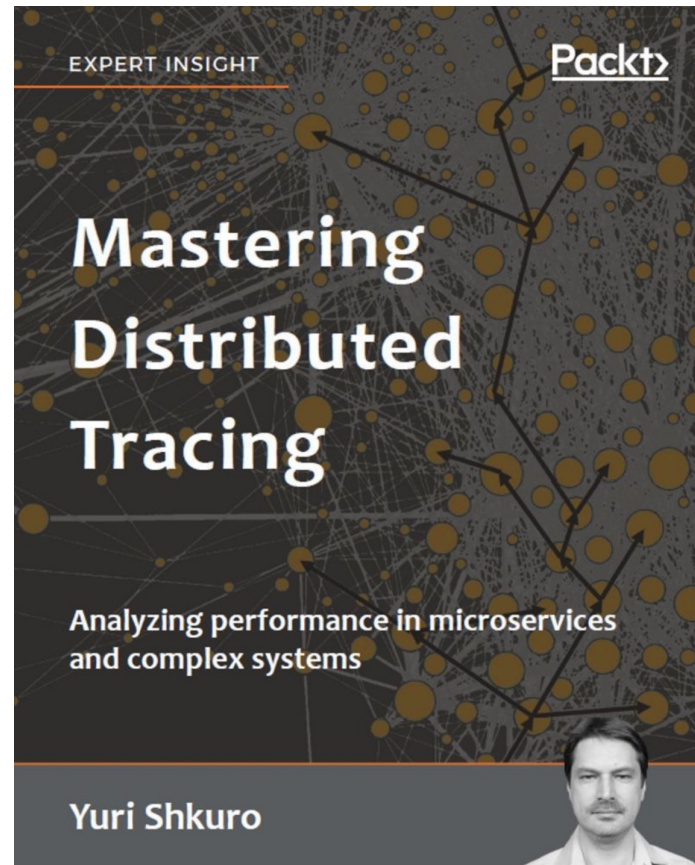
<https://www.youtube.com/watch?v=qg3yRmV-fHs>

Vostok Hercules: 3 года доставляем телеметрию  
— полёт нормальный  
(JPoint 2022)

Что почитать

# Что почитать

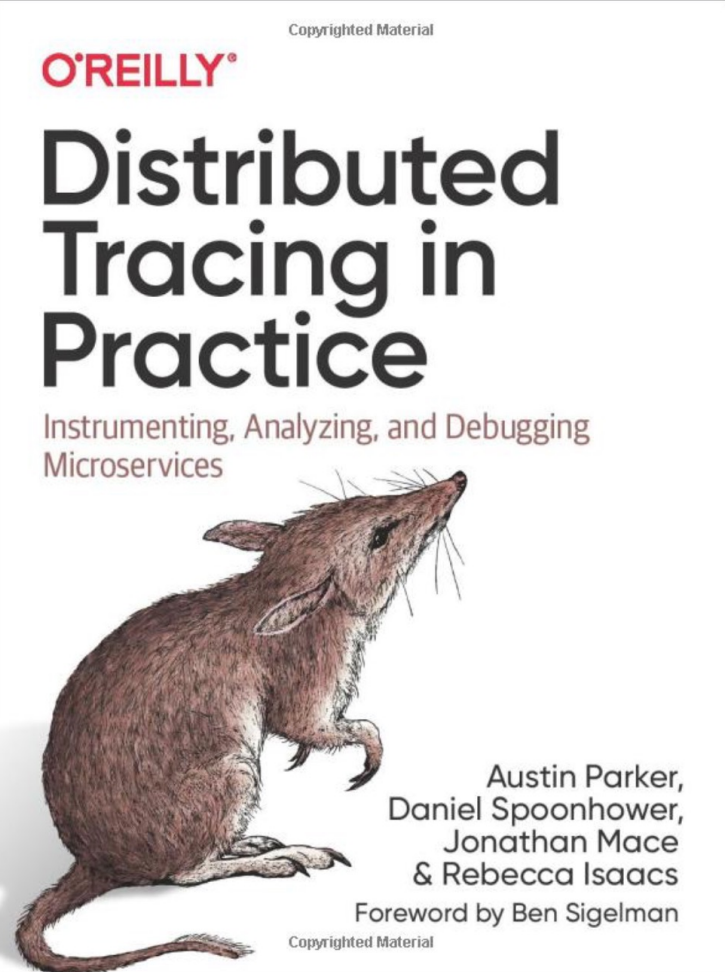
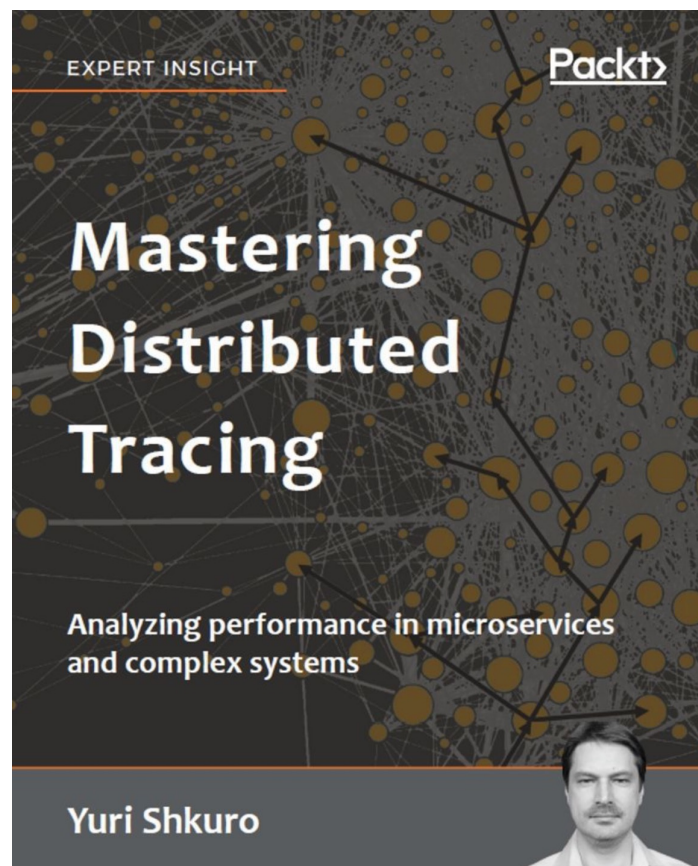
— Mastering Distributed Tracing, 2019



# Что почитать

— Mastering Distributed Tracing, 2019

— Distributed Tracing in Practice, 2020



У кого поучиться

У кого поучиться

— Uber (Jaeger)

# У кого поучиться

- Uber (Jaeger)
- Zipkin



# У кого поучиться

- Uber (Jaeger)
- Zipkin
- OpenTracing / OpenCensus → OpenTelemetry

# У КОГО ПОУЧИТЬСЯ

- Uber (Jaeger)
- Zipkin
- OpenTracing / OpenCensus → OpenTelemetry
- Lightstep
- New Relic

# У КОГО ПОУЧИТЬСЯ

- Uber (Jaeger)
- Zipkin
- OpenTracing / OpenCensus → OpenTelemetry
- Lightstep
- New Relic
- Google (Dapper)

# У КОГО ПОУЧИТЬСЯ

- Uber (Jaeger)
- Zipkin
- OpenTracing / OpenCensus → OpenTelemetry
- Lightstep
- New Relic
- Google (Dapper)
- AWS (X-Ray)

# У КОГО ПОУЧИТЬСЯ

- Uber (Jaeger)
- Zipkin
- OpenTracing / OpenCensus → OpenTelemetry
- Lightstep
- New Relic
- Google (Dapper)
- AWS (X-Ray)

...

# У КОГО ПОУЧИТЬСЯ

- Uber (Jaeger)
- Zipkin
- OpenTracing / OpenCensus → OpenTelemetry
- Lightstep
- New Relic
- Google (Dapper)
- AWS (X-Ray)

...

W3C Trace Context

# У КОГО ПОУЧИТЬСЯ

- Uber (Jaeger)
- Zipkin
- OpenTracing / OpenCensus → OpenTelemetry
- Lightstep
- New Relic
- Google (Dapper)
- AWS (X-Ray)
- ...

W3C Trace Context (Recommendation, 23 ноября 2021)

# У КОГО ПОУЧИТЬСЯ

- Uber (Jaeger)
- Zipkin
- OpenTracing / OpenCensus → OpenTelemetry
- Lightstep
- New Relic
- Google (Dapper)
- AWS (X-Ray)
- ...

W3C recommends the wide deployment of a Recommendation as a standard for the Web.

W3C Trace Context (Recommendation, 23 ноября 2021)



# Sampling — Семплирование

3 подхода:

- Обрабатывается весь поток спанов
- Трассировки семплируются в момент создания (head based sampling)
- Трассировки семплируются по завершению (tail based sampling)

# Sampling — Семплирование

3 подхода:

- **Обрабатывается весь поток спанов**
- Трассировки семплируются в момент создания  
(head based sampling)
- Трассировки семплируются по завершению  
(tail based sampling)

# Sampling — Семплирование

2.5 – 3 млн спанов / сек

# Sampling — Семплирование

2.5 – 3 млн спанов / сек

???

# Sampling — Семплирование

2.5 – 3 млн спанов / сек

??? = \$€₽

# Инструментация

3 подхода:

- Black-box
- В коде
- Agent-based (модификация исполняемого кода)

# Инструментация

3 подхода:

— Black-box

— **В коде**

— Agent-based (модификация исполняемого кода)

# Инструментация

3 подхода:

- Black-box
- **В коде**
- Agent-based (модификация исполняемого кода)

А как же Service Mesh?



# Инструментация

3 подхода:

- Black-box
- **В коде**
- Agent-based (модификация исполняемого кода)

А как же Service Mesh?

Распределённая трассировка не работает без инструментации внутри приложения (копирование заголовков)

Зачем нужны трассировки



**Honest Update**  
@honest\_update



We replaced our monolith with micro services so that every outage could be more like a murder mystery.

4:10 AM · 8 окт. 2015 г. · Buffer

---

**2 900** ретвитов   **73** твита с цитатами   **2 709** отметок «Нравится»

# 5 задач трассировок

- Мониторинг распределённых транзакций
- Оптимизация производительности (latency)
- Поиск причин (ошибок, деградации)
- Анализ зависимости сервисов
- Распространение контекста

# 5 задач трассировок

- Мониторинг распределённых транзакций
- Оптимизация производительности (latency)
- Поиск причин (ошибок, деградации)
- Анализ зависимости сервисов
- **Распространение контекста**

# Тегирование трафика

Когда полезно навешивать ярлыки

# Тегирование трафика

Разметка спанов различными тегами

# Тегирование трафика

Разметка спанов различными тегами

— ДЦ, FQDN, host



# Тегирование трафика

Разметка спанов различными тегами

- ДЦ, FQDN, host
- Продукт

# Тегирование трафика

Разметка спанов различными тегами

- ДЦ, FQDN, host
- Продукт
- Клиент / пользователь

# Тегирование трафика

Разметка спанов различными тегами

- ДЦ, FQDN, host
- Продукт
- Клиент / пользователь

...

# Тегирование трафика

Консистентность

# Тегирование трафика

## Консистентность

— Конвенция по именованию тегов (operation, service, version, ...)

# Тегирование трафика

## Консистентность

- Конвенция по именованию тегов (operation, service, version, ...)
- Должно учитывать размерность и тип данных

# Тегирование трафика

## Консистентность

- Конвенция по именованию тегов (operation, service, version, ...)
- Должно учитывать размерность и тип данных

Во всех системах!

# Диагностика на основе тегов



# Диагностика на основе тегов

— Долгие запросы из-за повторных попыток?  
(retry = true)

# Диагностика на основе тегов

— Долгие запросы из-за повторных попыток?  
(retry = true)

— Отсутствие данных в кэше даёт основной вклад в latency?  
(cache = miss)

# Диагностика на основе тегов

— Долгие запросы из-за повторных попыток?  
(retry = true)

— Отсутствие данных в кэше даёт основной вклад в latency?  
(cache = miss)

— Долгие запросы из-за промежуточных таймаутов?  
(status code = 499)

# Тестирование на проде

# Тестирование на проде

Специальный тег для тестовых аккаунтов

# Тестирование на проде

Специальный тег для тестовых аккаунтов

— Автоматический перевод в read-only режим

# Тестирование на проде

Специальный тег для тестовых аккаунтов

- Автоматический перевод в read-only режим
- Отправлять синтетический трафик на staging при рутинге

# Тестирование на проде

Специальный тег для тестовых аккаунтов

- Автоматический перевод в read-only режим
- Отправлять синтетический трафик на staging при роутинге
- Multi-tenancy: не паниковать при росте нагрузки из-за тестов



# Chaos Engineering

# Chaos Engineering

Теги со специальными инструкциями:

# Chaos Engineering

Теги со специальными инструкциями:

— увеличить latency

# Chaos Engineering

Теги со специальными инструкциями:

- увеличить latency
- завершиться с timeout

# Chaos Engineering

Теги со специальными инструкциями:

- увеличить latency
- завершиться с timeout
- завершиться с определённой ошибкой

# Chaos Engineering

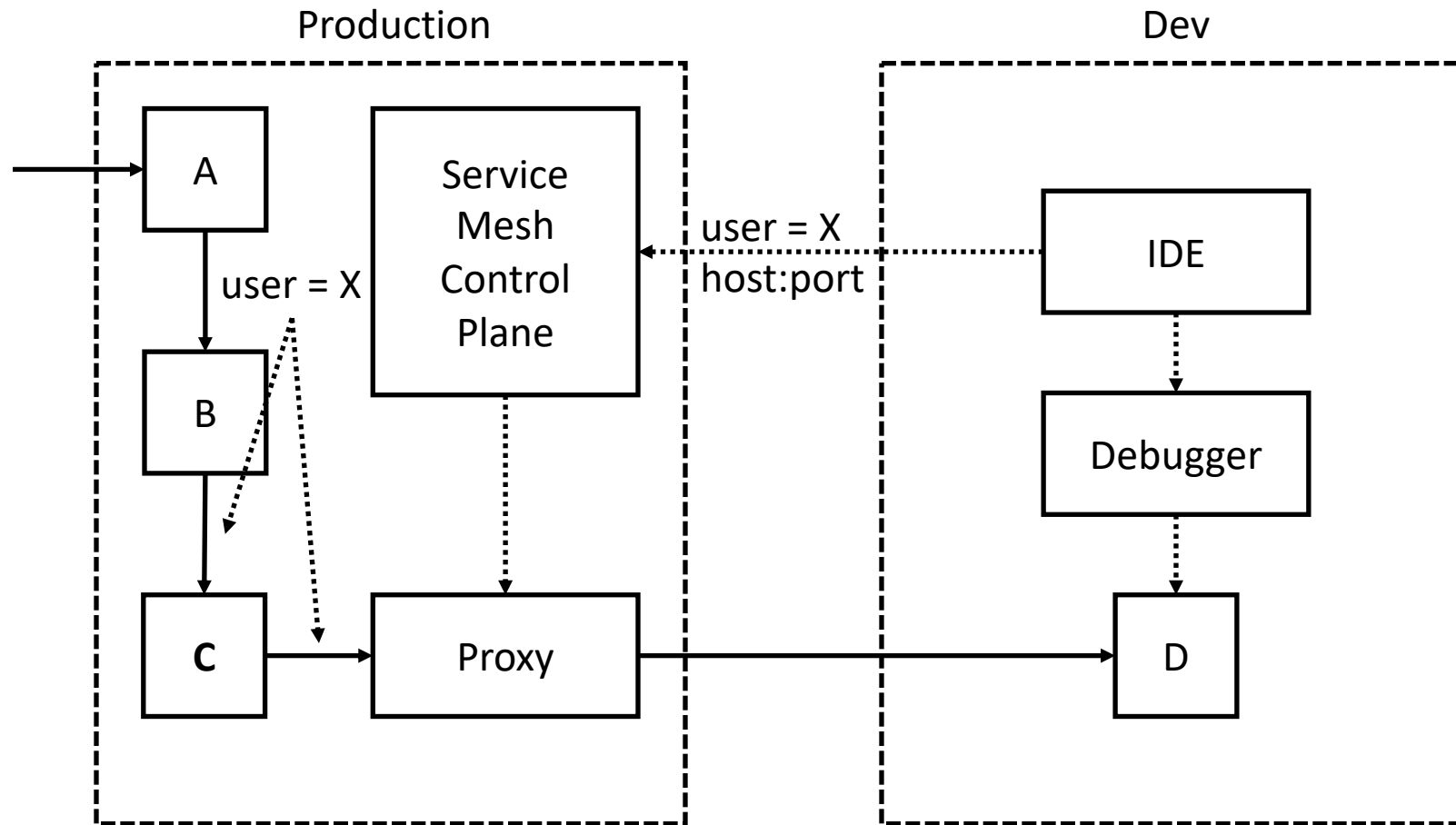
Теги со специальными инструкциями:

- увеличить latency
- завершиться с timeout
- завершиться с определённой ошибкой
- ...

# Отладка на проде

Squash Debugger

# Разработка в проде





# Поиск по тегам

# Поиск по тегам

Zipkin Investigate system behavior Find a trace View Saved Trace Dependencies

Go to trace

Service Name

all

Span Name

all

Lookback

1 hour

Annotations Query

e.g. "http.path=/foo/bar/ and cluster=foo and cache.miss"

Find Traces

?

Duration (µs) >=

Limit

10

Sort

Longest First

Please select the criteria for your trace lookup.

# Поиск по тегам

Zipkin Investigate system behavior Find a trace View Saved Trace Dependencies

Go to trace

Service Name

all

Span Name

all

Lookback

1 hour

Annotations Query

e.g. "http.path=/foo/bar/ and cluster=foo and cache.miss"

Find Traces

?

Duration (µs) >=

Limit

10

Sort

Longest First

Please select the criteria for your trace lookup.

# Поиск по тегам

В дополнение к тегам

- Длительность запроса (трассировки)
- Период времени

# «Интересные» трассировки

# «Интересные» трассировки

— Ошибки (HTTP: 4xx, 5xx)

# «Интересные» трассировки

- Ошибки (HTTP: 4xx, 5xx)
- Долгие запросы (время выполнения > 1 сек)

# «Интересные» трассировки

Zipkin Investigate system behavior Find a trace View Saved Trace Dependencies

Go to trace

Service Name

all

Span Name

all

Lookback

1 hour

Annotations Query

e.g. "http.path=/foo/bar/ and cluster=foo and cache.miss"

Duration (µs) >=

Limit

10

Sort

Longest First

Find Traces

?

Please select the criteria for your trace lookup.



# Интеграция с логами

# Интеграция с логами

— Поиск логов по trace id

# Интеграция с логами

- Поиск логов по trace id
- Поиск логов по span id

# Интеграция с логами

- Поиск логов по trace id
- Поиск логов по span id

В каждую лог-запись добавляем trace id и span id

# Интеграция с логами

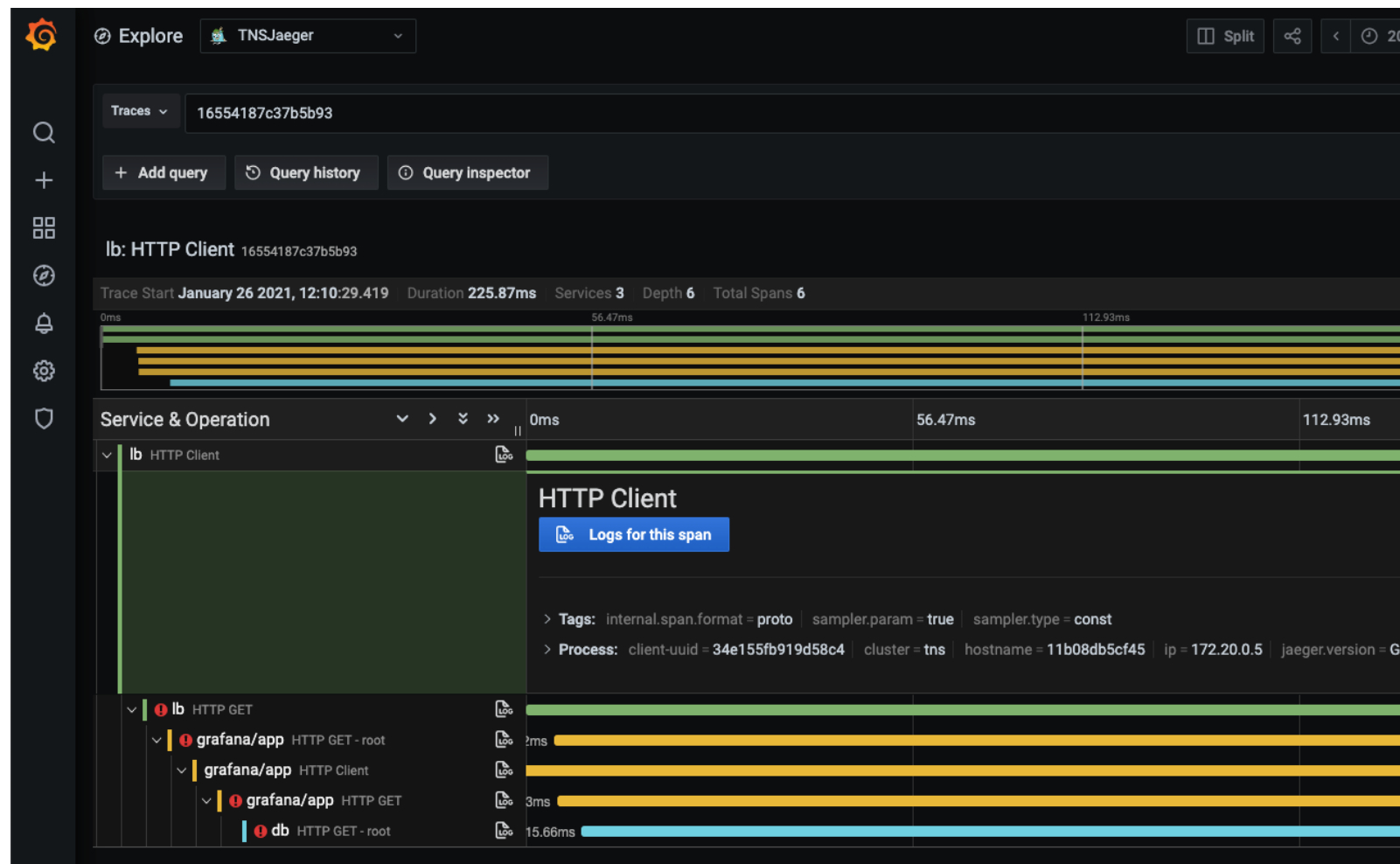
- Поиск логов по trace id
- Поиск логов по span id

В каждую лог-запись добавляем trace id и span id

Просмотр логов  
в системе с трассировками

# Интеграция с логами

Grafana  
— Tempo  
— Jaeger  
— Zipkin



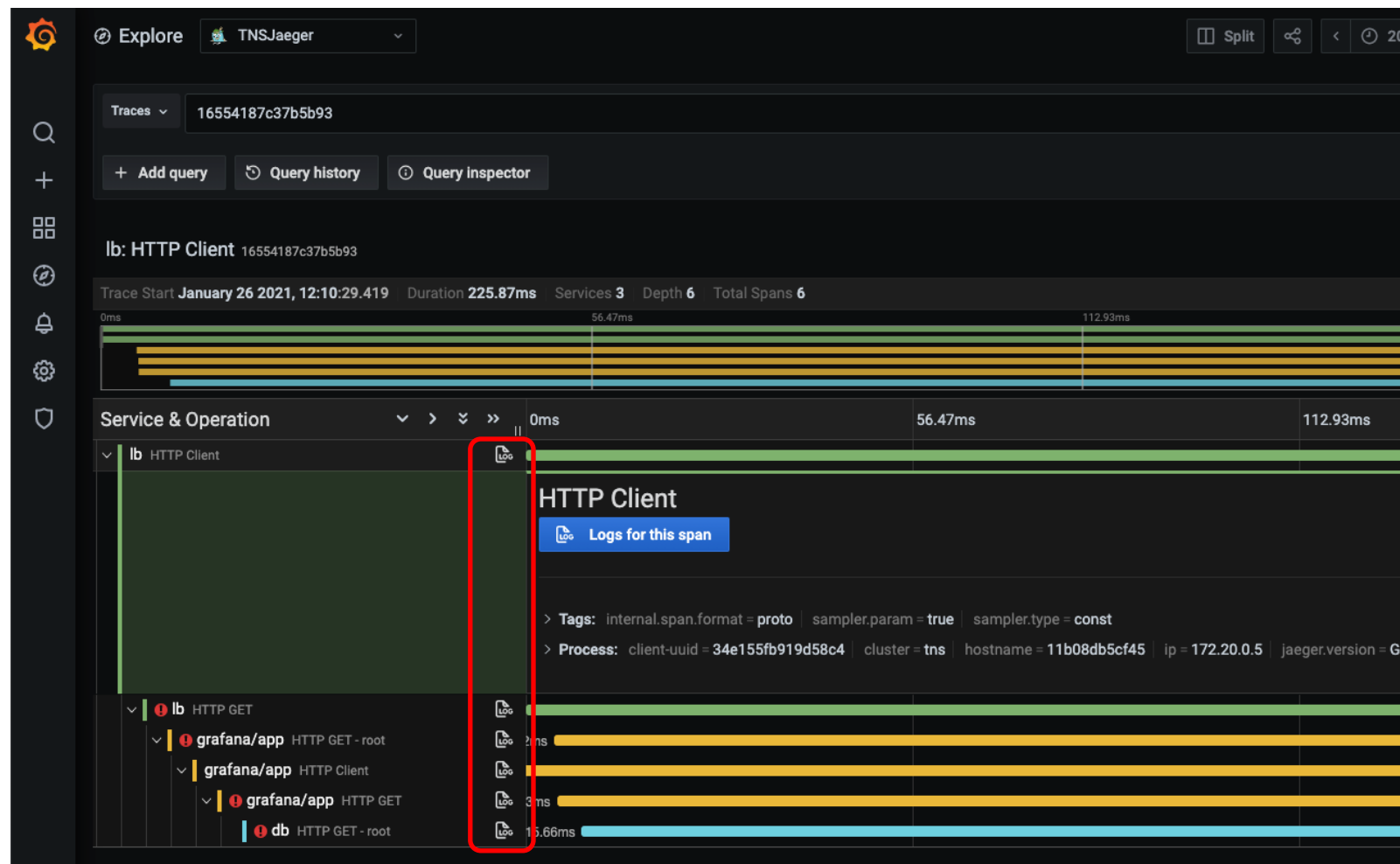
# Интеграция с логами

Grafana

— Tempo

— Jaeger

— Zipkin



Логи не нужны



# Логи не нужны

events в OpenTelemetry

logs в OpenTracing

Трассировки — инструмент диагностики

Трассировки — инструмент диагностики

Метрики — инструмент обнаружения

# Трассировки → Метрики

Трассировки — инструмент диагностики

Метрики — инструмент обнаружения

# Метрики

# Метрики

— 4 Golden Signals (latency, traffic, errors, saturation)

# Метрики

— 4 Golden Signals (latency, traffic, errors, saturation)  
RED (rate, error, duration)

# Метрики

— 4 Golden Signals (latency, traffic, errors, saturation)  
RED (rate, error, duration)

Стандартные теги у спана

- время начала и завершения
- размер тела запроса и ответа
- код ответа



# Метрики

- 4 Golden Signals (latency, traffic, errors, saturation)  
RED (rate, error, duration)

- Добавление контекста к метрикам  
(тегированные метрики)

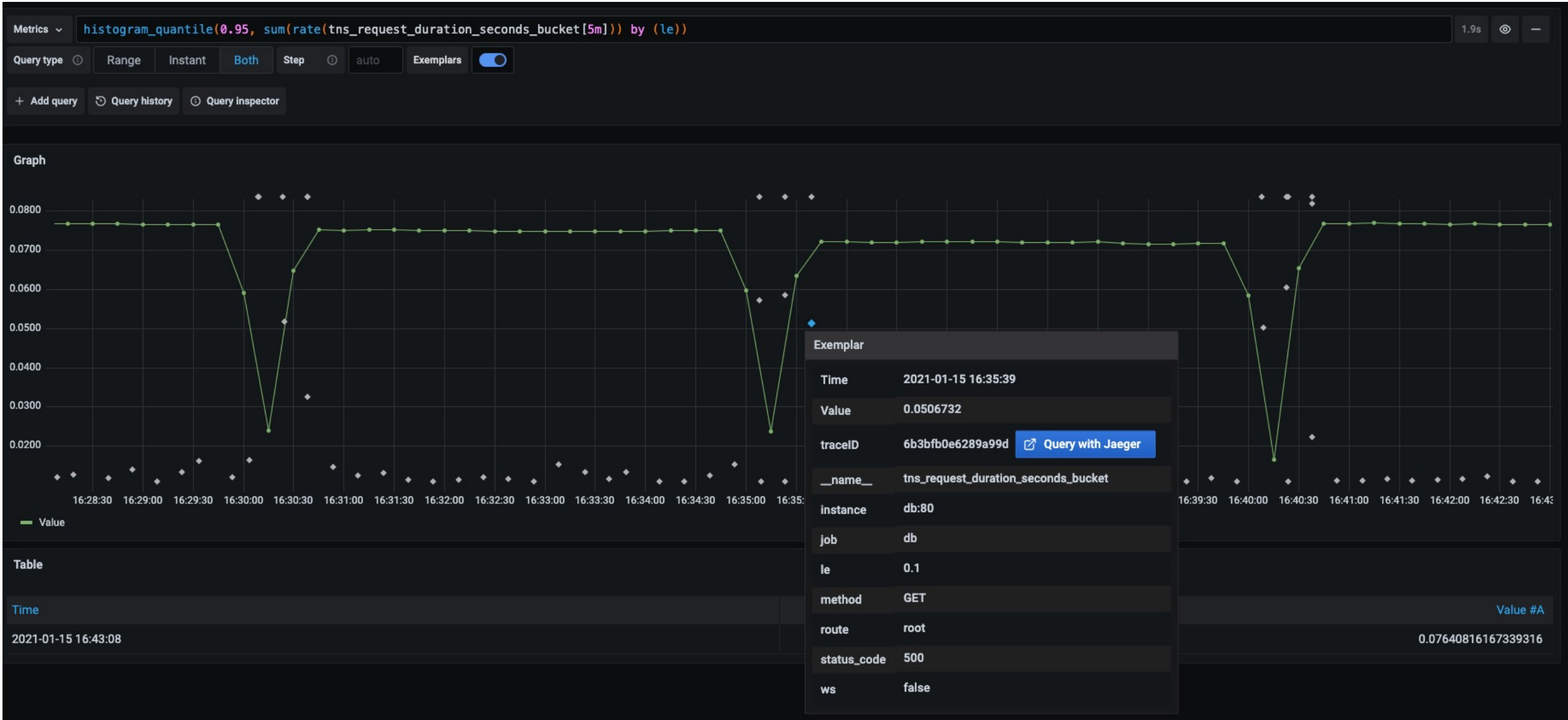
# Метрики

- 4 Golden Signals (latency, traffic, errors, saturation)  
RED (rate, error, duration)

- Добавление контекста к метрикам  
(тегированные метрики)

К метрике request latency добавляем теги

- версию сервиса
- хост / ДЦ



Мат.статистика

# Гистограмма latency

# Гистограмма latency

Различные performance-профили в зависимости от параметров запроса

# Гистограмма latency

Различные performance-профили в зависимости от параметров запроса



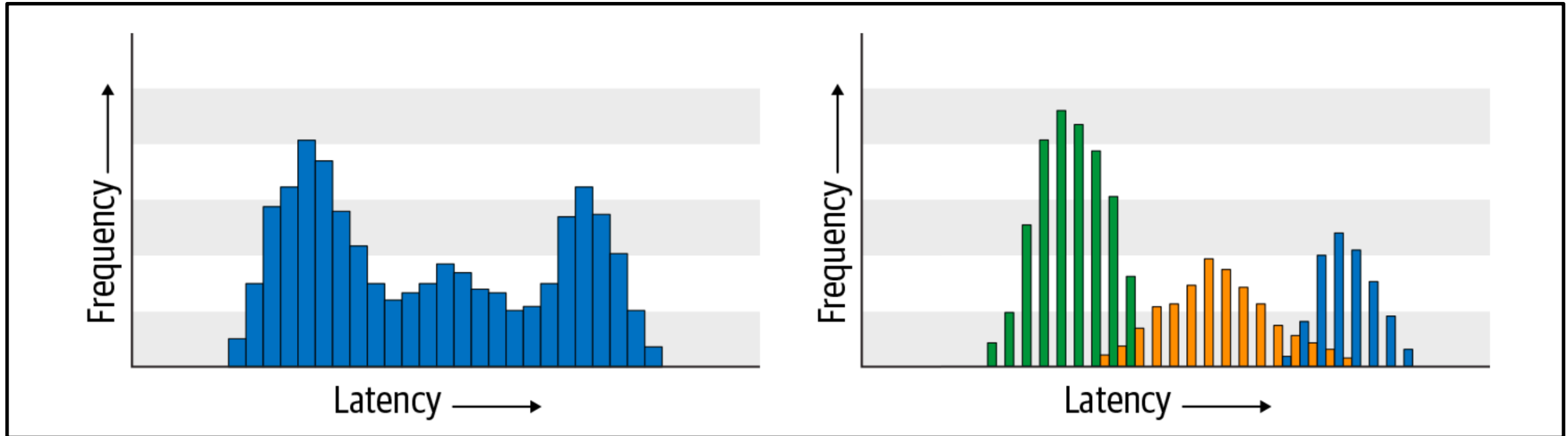
# Гистограмма latency

Различные performance-профили в зависимости от параметров запроса

— Нужно смотреть на распределение



# Гистограмма latency



# Гистограмма latency

Различные performance-профили в зависимости от параметров запроса

- Нужно смотреть на распределение
- Сравнение запросов из разных модальностей

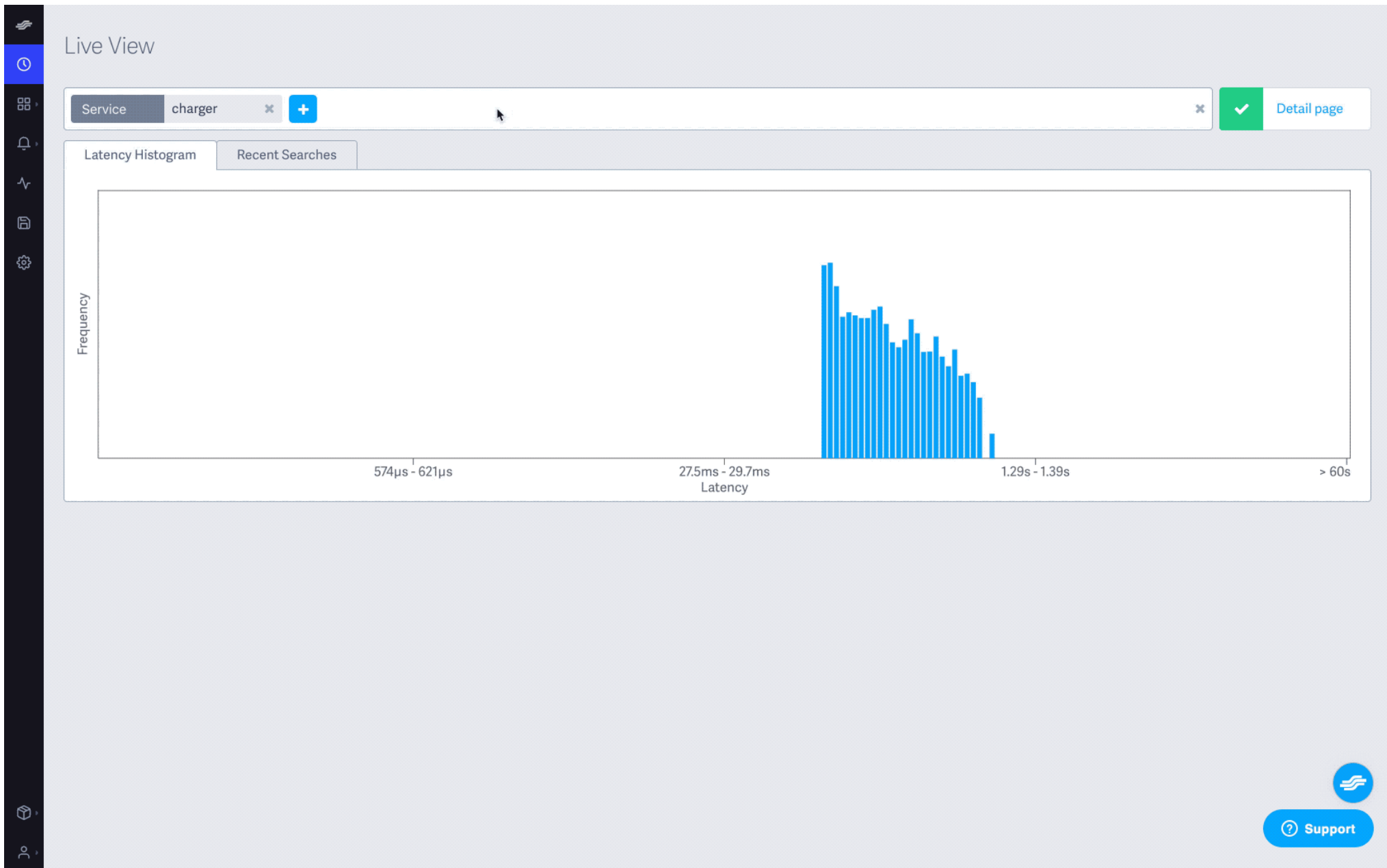
# Гистограмма latency

Различные performance-профили в зависимости от параметров запроса

- Нужно смотреть на распределение
- Сравнение запросов из разных модальностей

Шкала бакетов

- равномерная
- прогрессивная
- задаётся вручную



# Тренды

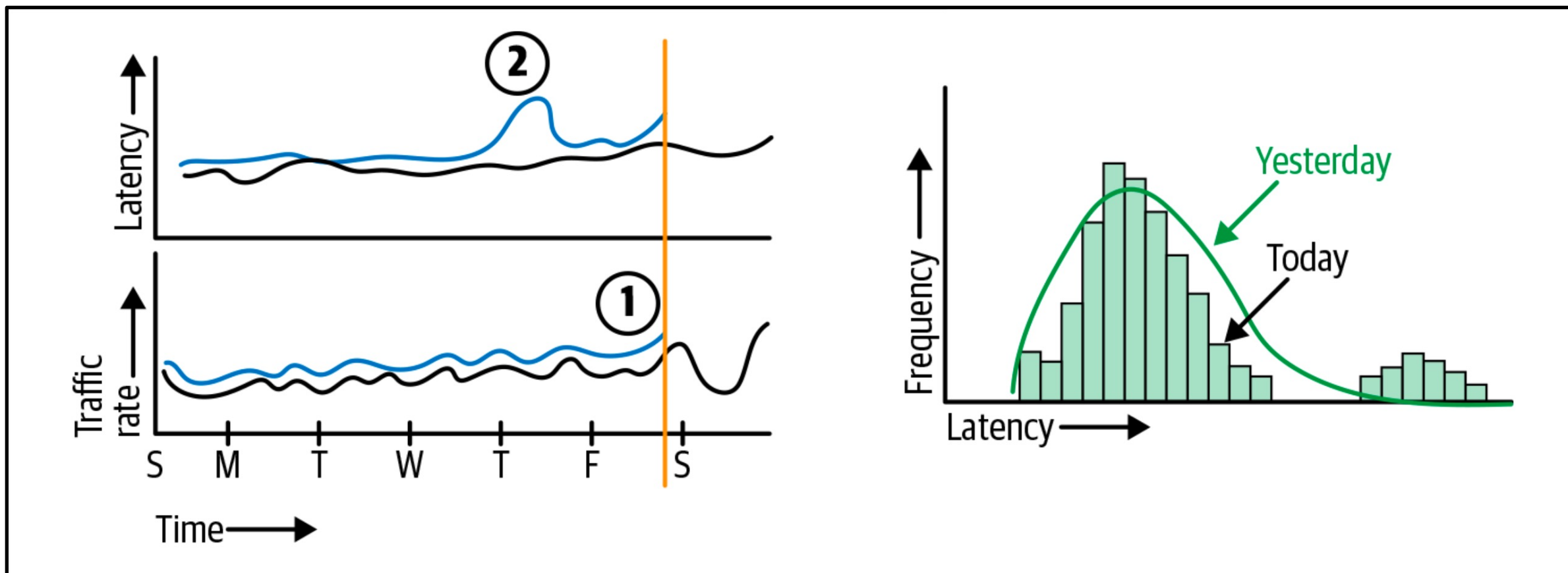
# Тренды

Цель: увидеть деградацию как можно раньше

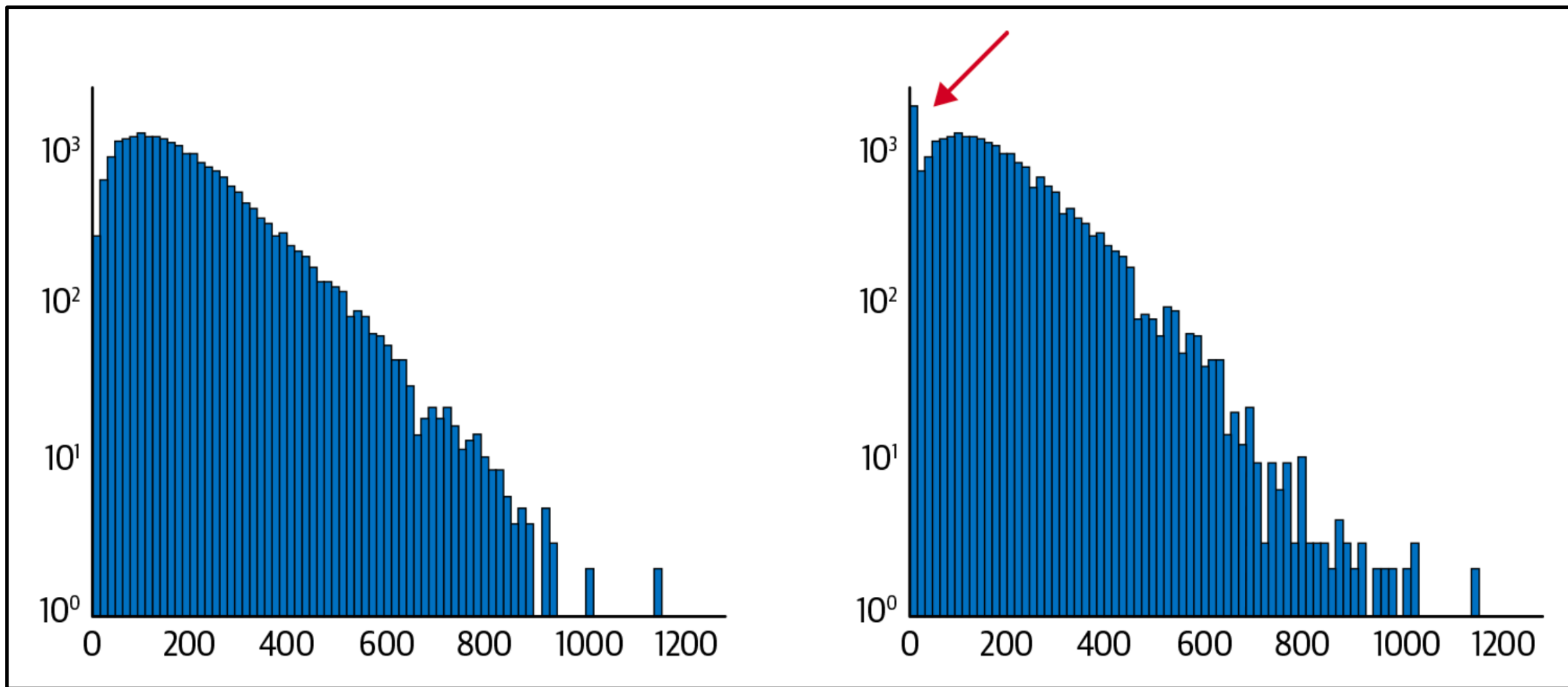
— p99.9 latency сейчас и месяц назад

— Гистограмма latency сейчас и месяц назад

# Тренды



# Тренды





# Анализ корреляции

# Анализ корреляции

Выявление корреляции между latency и тегами

# Анализ корреляции

Выявление корреляции между latency и тегами

Получение гипотез о деградации  
на основе мат.статистики

# Визуализация трассировки

# Critical path — Критический путь

# Critical path — Критический путь

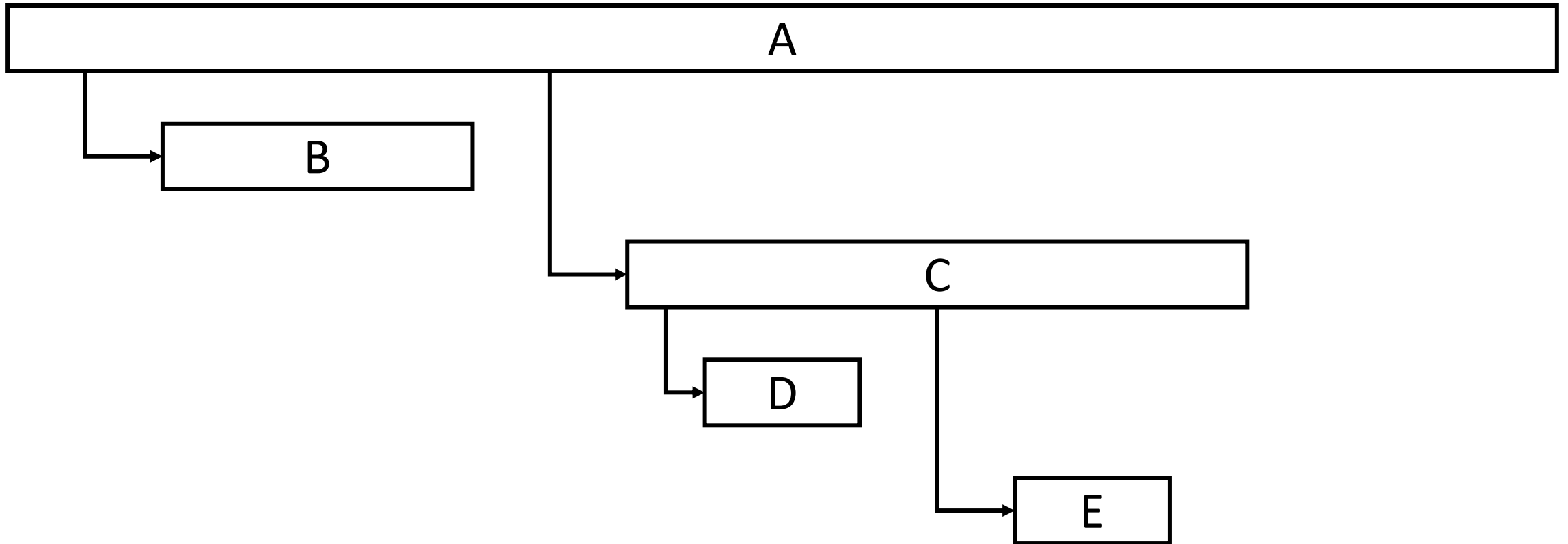
Critical path — последовательность задач проекта, имеющих нулевой резерв времени выполнения

# Critical path — Критический путь

Critical path — последовательность задач проекта, имеющих нулевой резерв времени выполнения

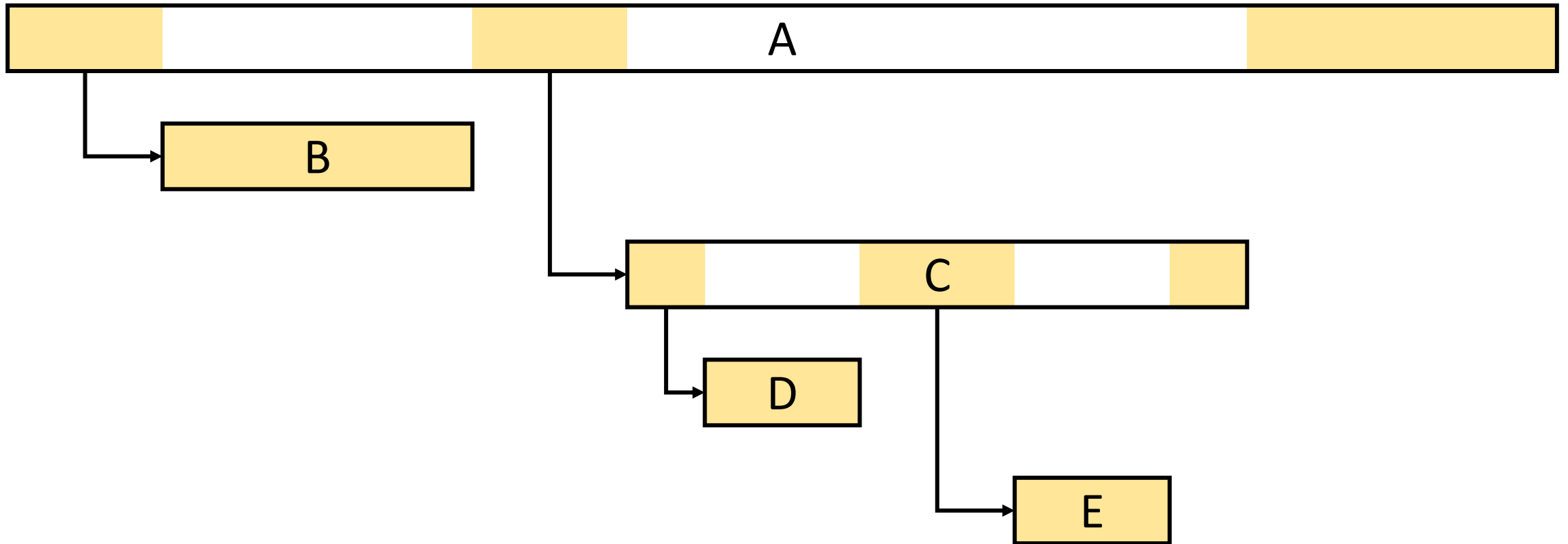
Если изменить длительность задачи, изменится срок проекта

# Critical path — Критический путь

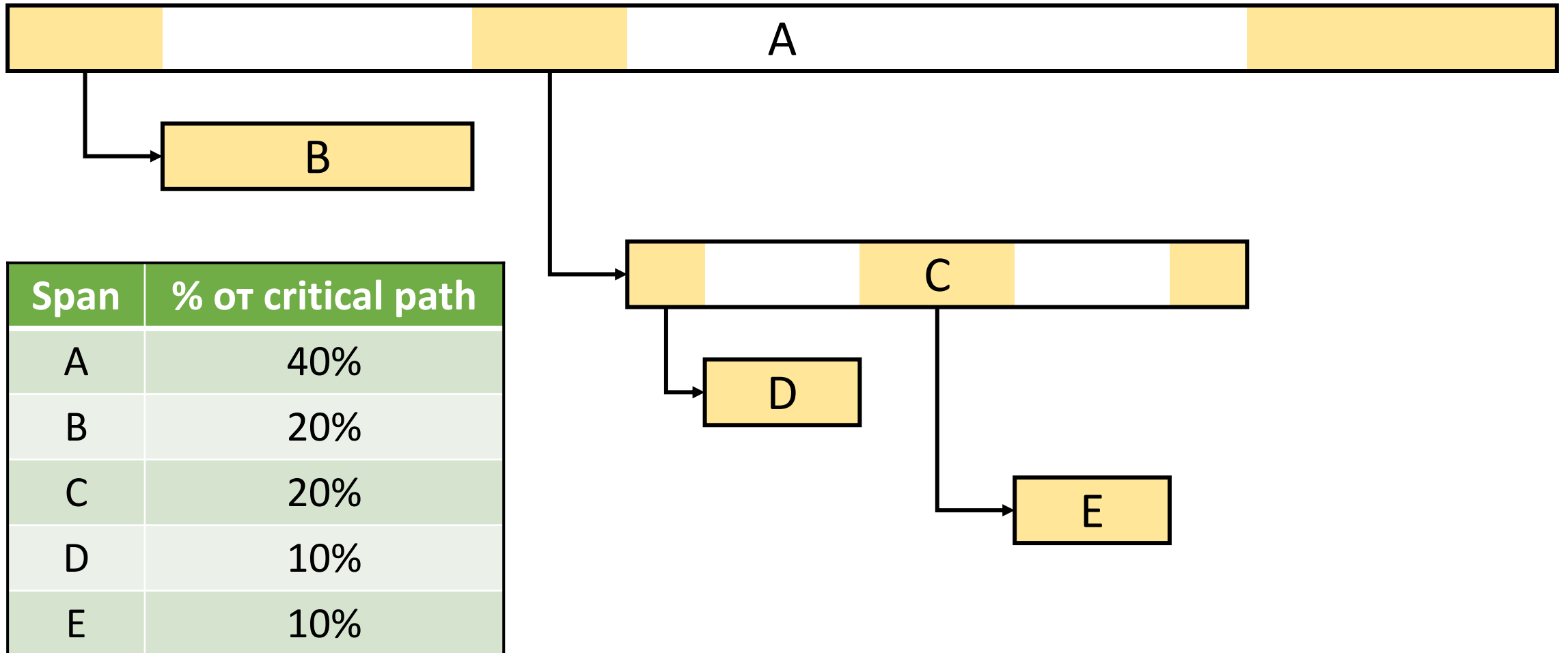




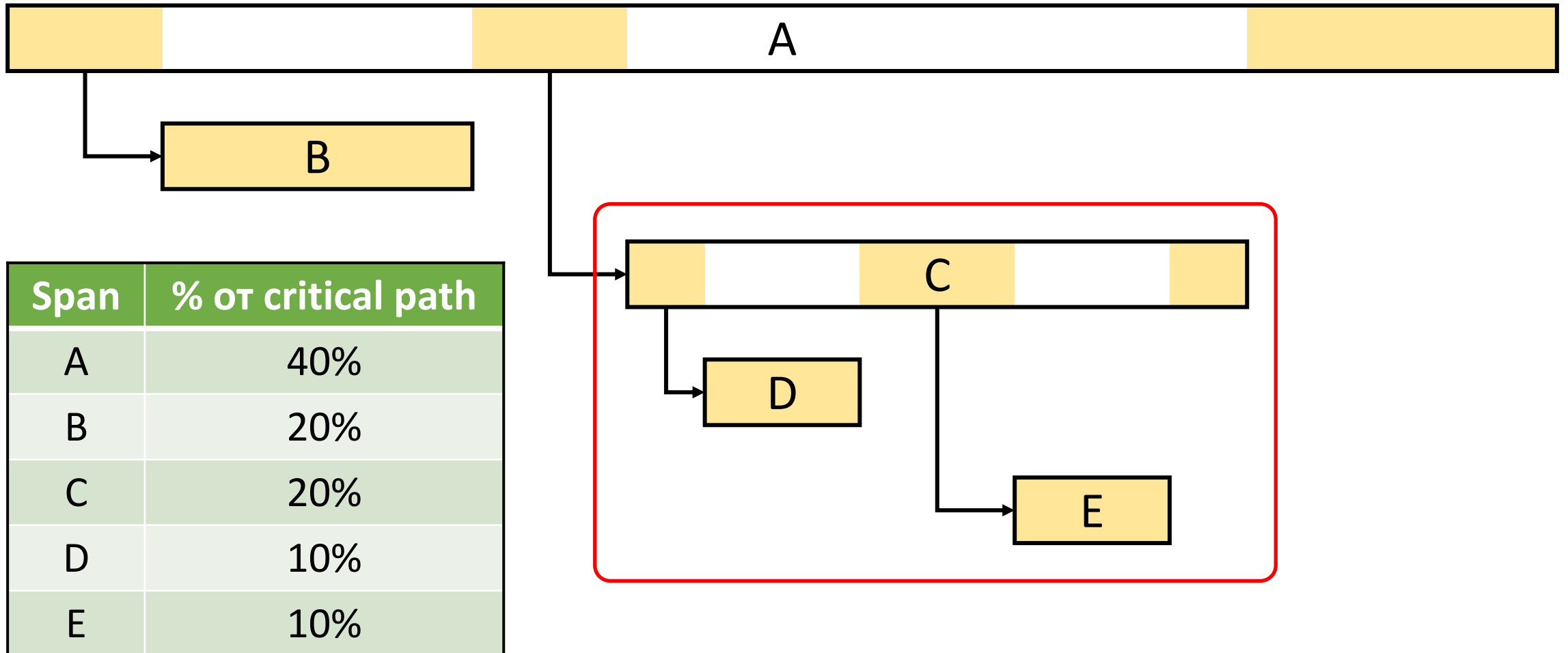
# Critical path — Критический путь



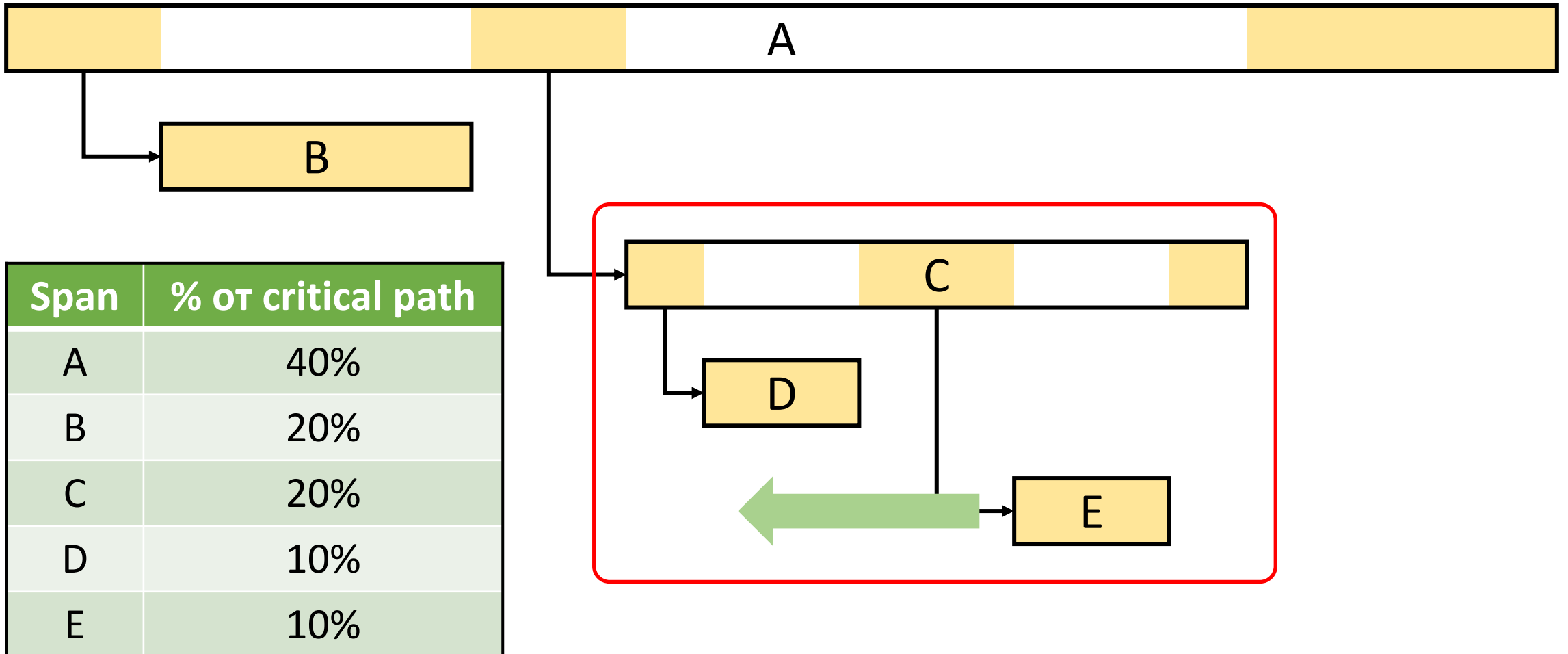
# Critical path — Критический путь



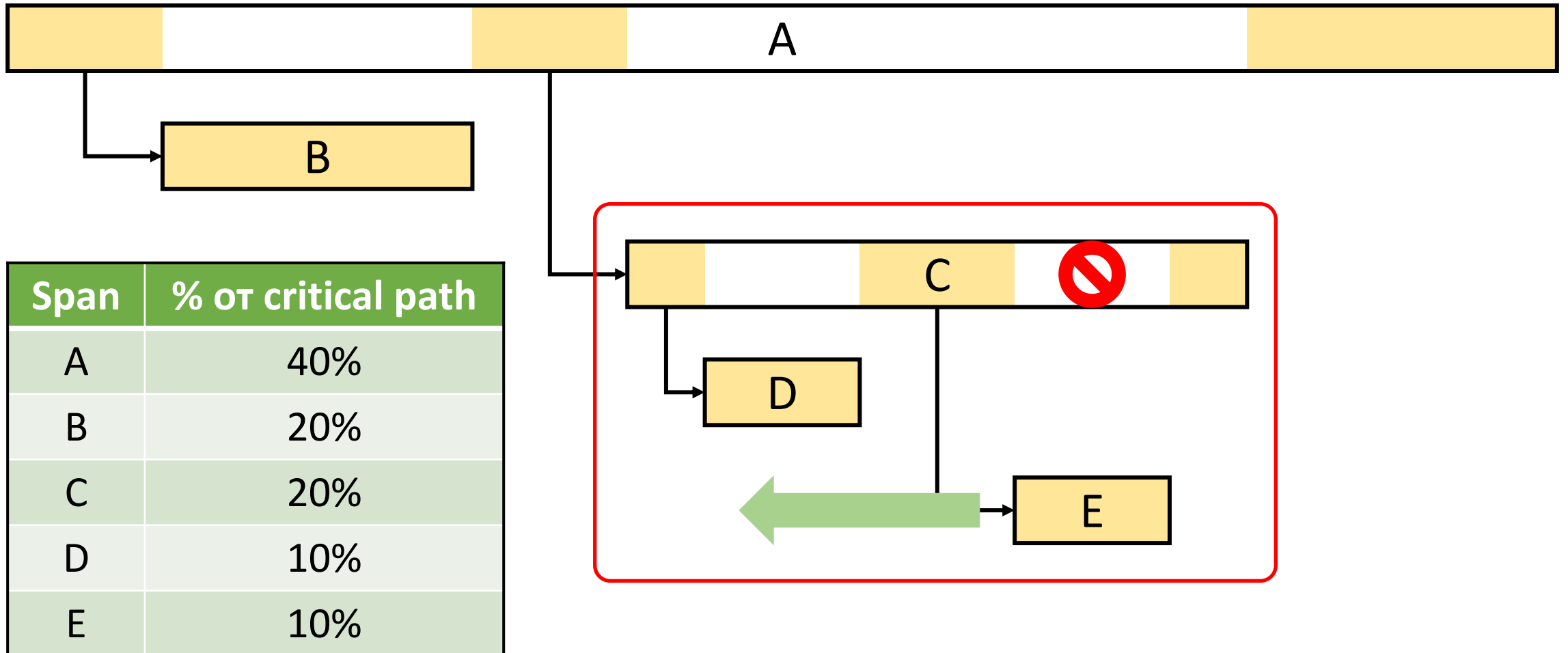
# Critical path — Критический путь



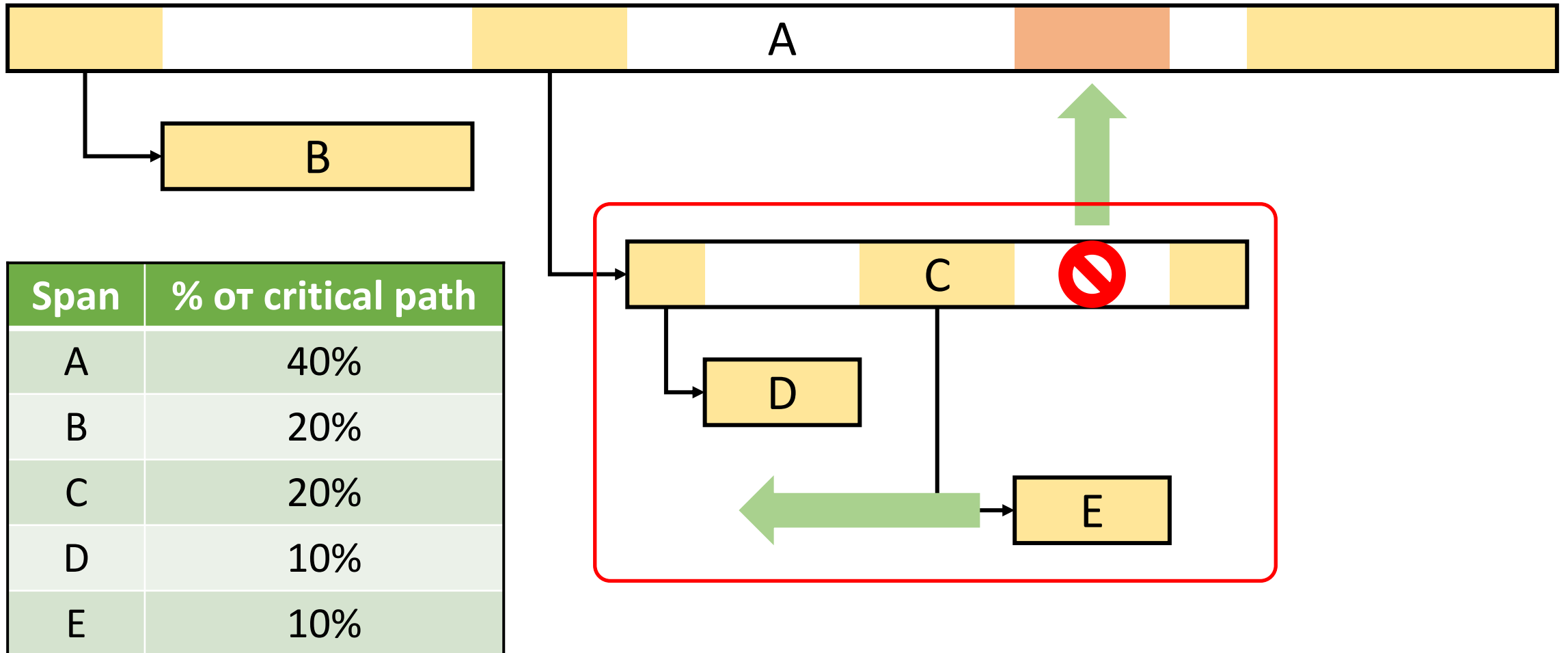
# Critical path — Критический путь



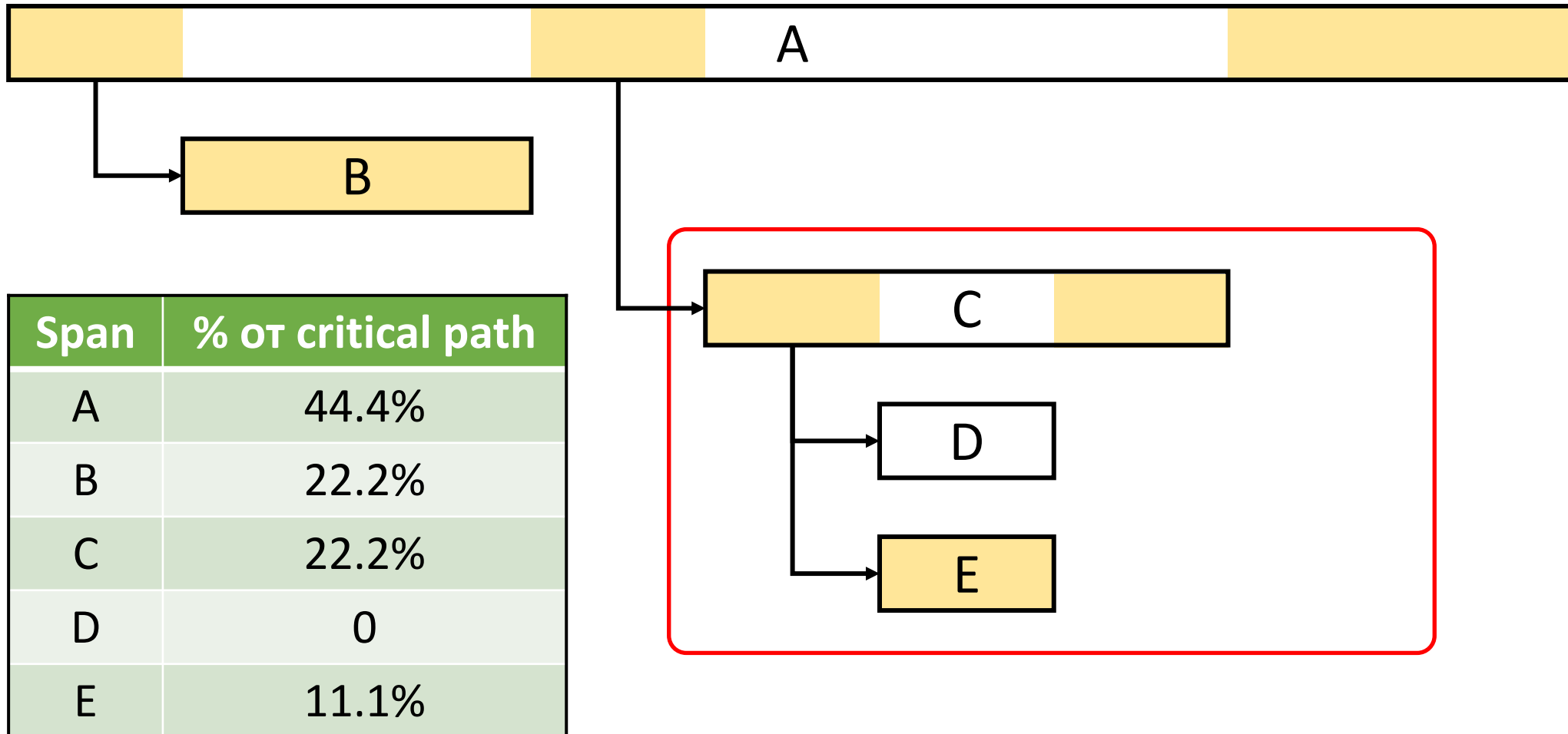
# Critical path — Критический путь



# Critical path — Критический путь



# Critical path — Критический путь



# Паттерны трассировок

- Маркер «ошибка»
- Самый длинный спан в critical path
- «Дыры» в инструментации

Пример: запрос к БД, который не инструментируется

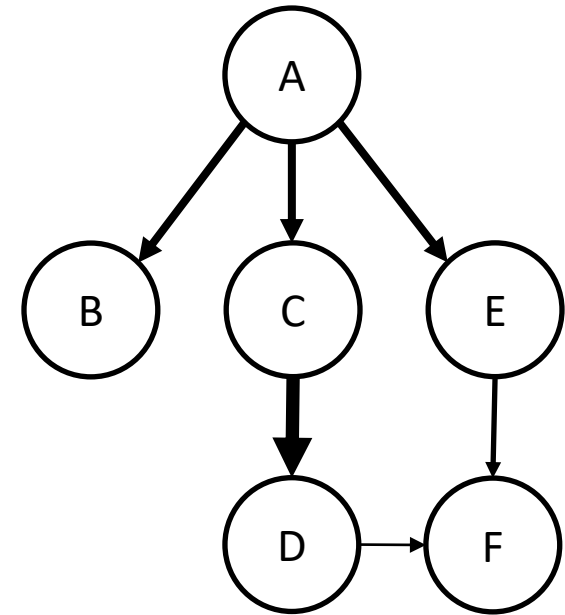
- Последовательное выполнение («лесенка»)
  - Несколько спанов завершается в один момент времени
- Пример: lock contention (транзакция в БД)
- Пример: timeout + cancellation



# Сервисный граф

# Сервисный граф

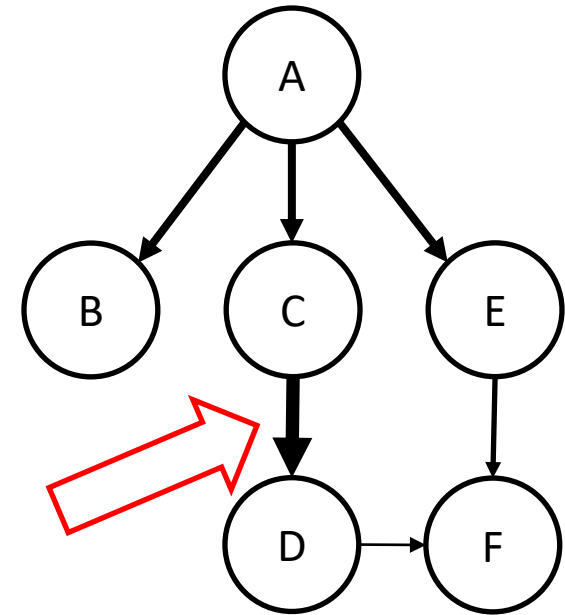
На всём множестве спанов



# Сервисный граф

На всём множестве спанов

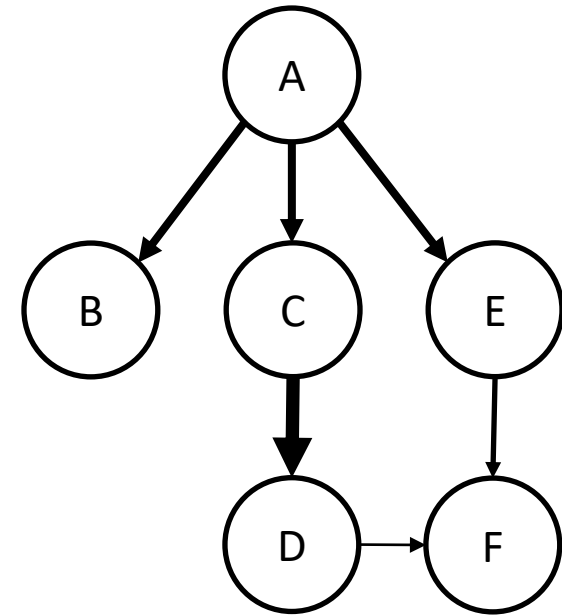
— Увидеть места, где много запросов



# Сервисный граф

На всём множестве спанов

- Увидеть места, где много запросов
  - Анализ зависимости сервисов
- Есть ли реальная связь между сервисами  
(по эмпирическим данным)

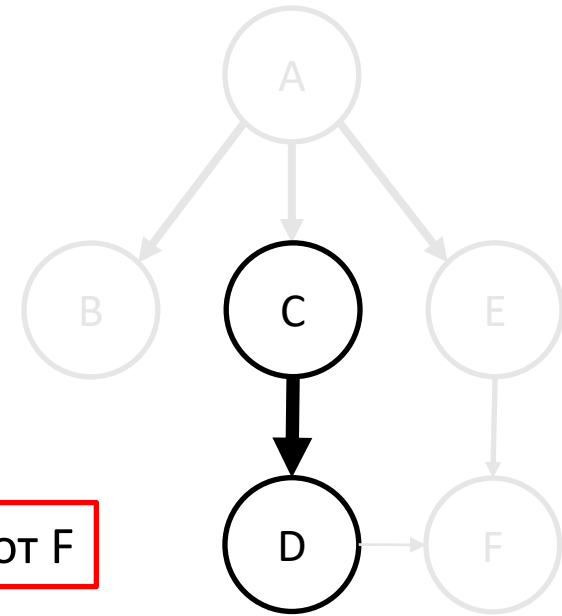


# Сервисный граф

На всём множестве спанов

- Увидеть места, где много запросов
  - Анализ зависимости сервисов
- Есть ли реальная связь между сервисами  
(по эмпирическим данным)

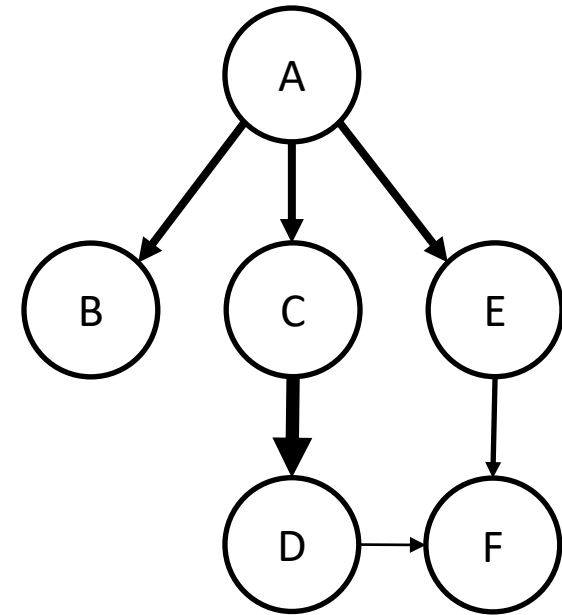
Сервис C не зависит от F



# Сервисный граф

На всём множестве спанов

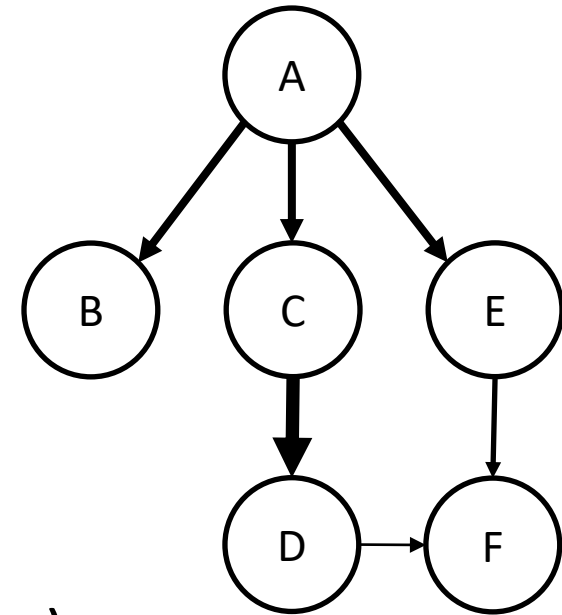
- Увидеть места, где много запросов
  - Анализ зависимости сервисов
- Есть ли реальная связь между сервисами  
(по эмпирическим данным)
- Фильтровать по тегам

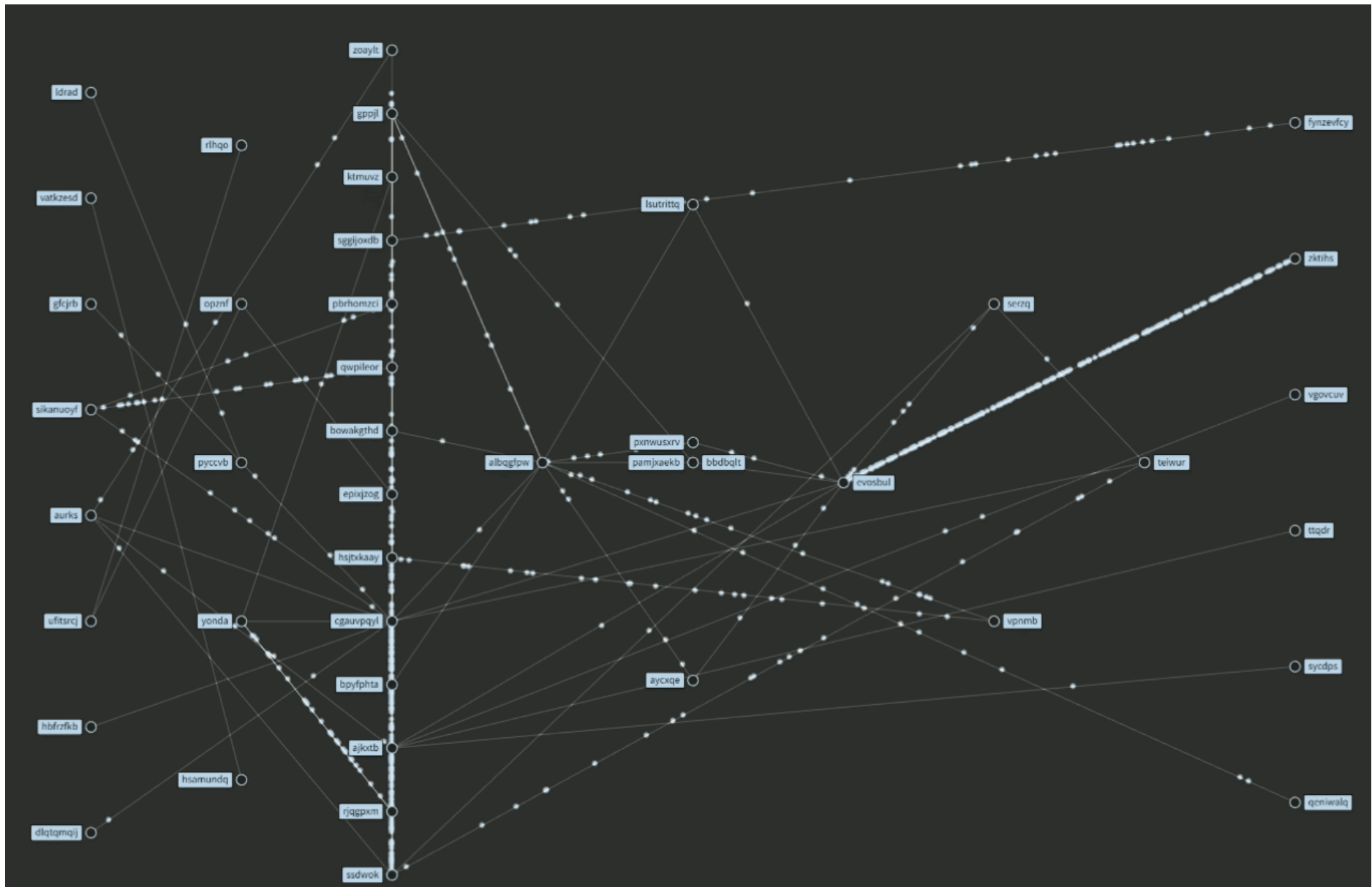


# Сервисный граф

На всём множестве спанов

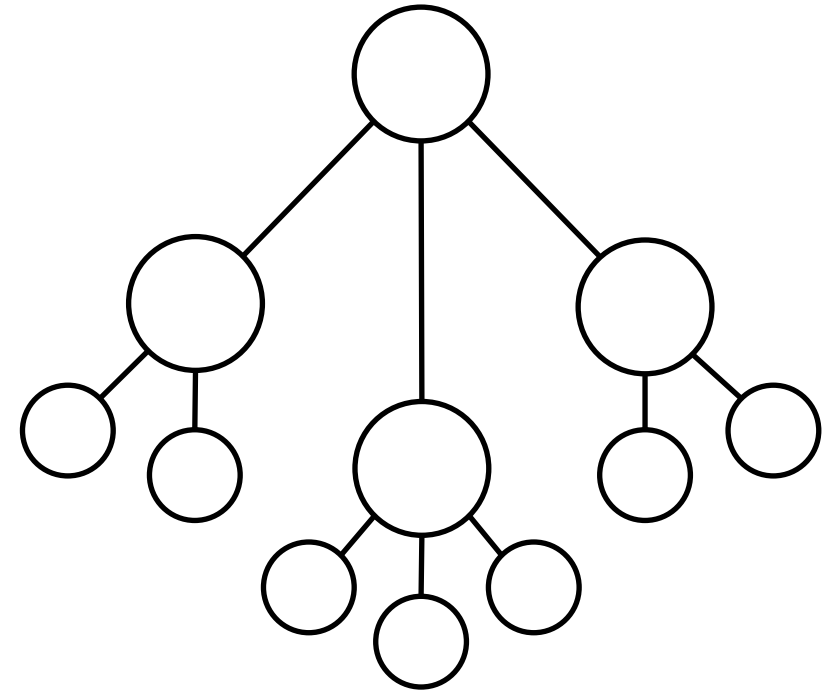
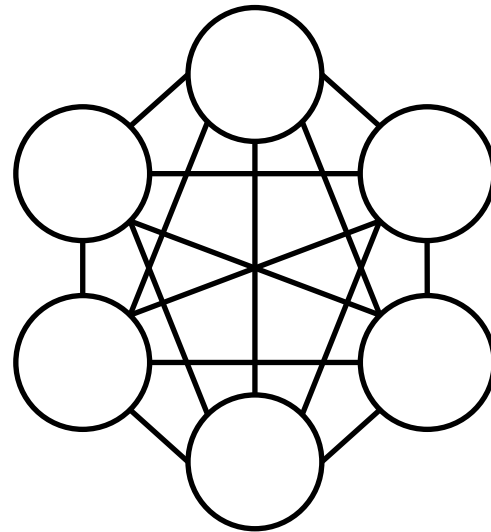
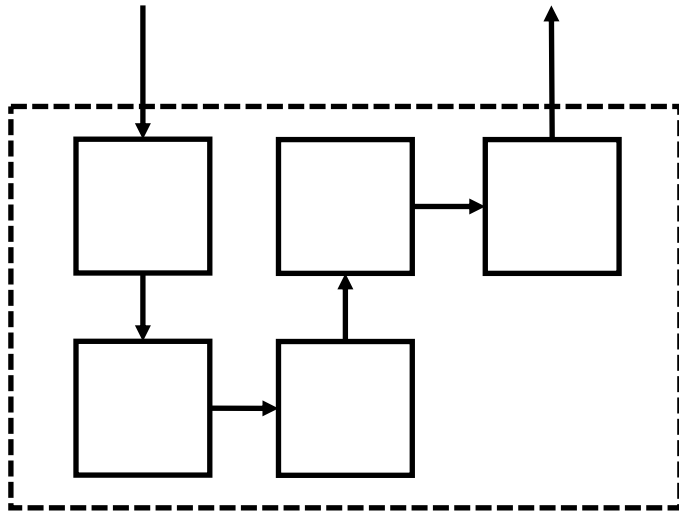
- Увидеть места, где много запросов
  - Анализ зависимости сервисов
- Есть ли реальная связь между сервисами (по эмпирическим данным)
- Фильтровать по тегам
  - Дополнить метриками (RPS, latency pX, EPS, ...)







# Обнаружение архитектурных проблем



# Биллинг

Фиксация утилизации ресурсов в разрезе тегов

# Биллинг

Фиксация утилизации ресурсов по нужному тегу

# Биллинг

Фиксация утилизации ресурсов по нужному тегу

- организация / пользователь
- сессия / подписка
- root-сервис

# Биллинг

Сервис X прирастёт на 30%

# Биллинг

Сервис X прирастёт на 30%

— Планирование ресурсов

# Биллинг

Сервис X прирастёт на 30%

- Планирование ресурсов
- Затраты на сервис в инфраструктуре

# Биллинг

Сервис X прирастёт на 30%

- Планирование ресурсов
- Затраты на сервис в ~~инфраструктуре~~ любом другом сервисе

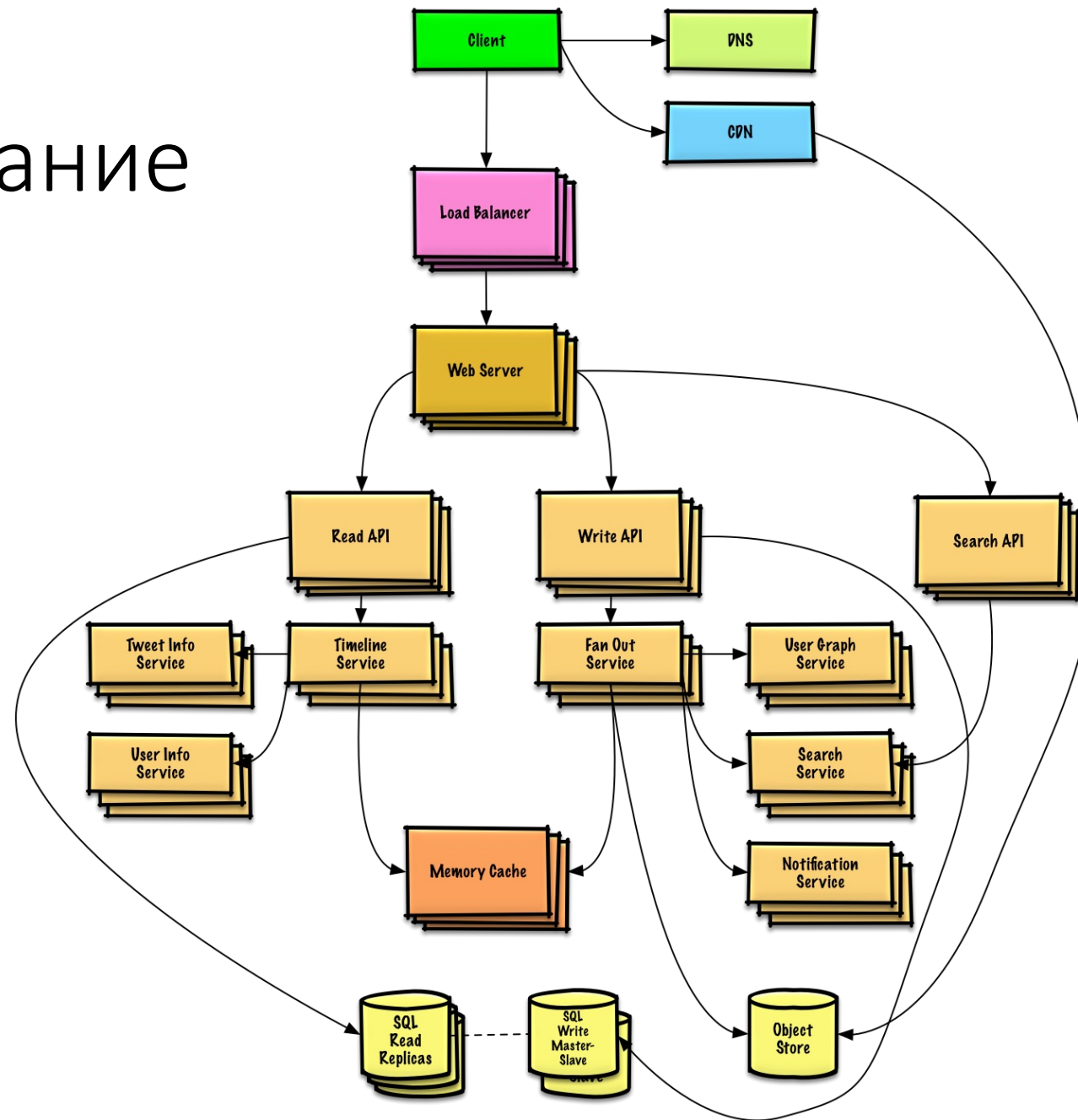


T  
D  
D

# Trace Driven Development

# Проектирование

# Проектирование



# Тестирование

# Тестирование

Описания тестового сценария на языке спанов

# Launch checklist

# Launch checklist

Чеклист для запуска сервиса



# Launch checklist

Чеклист для запуска сервиса

...

— Настроить Health Check

# Launch checklist

Чеклист для запуска сервиса

...

- Настроить Health Check
- **Сделать тестовый запрос и сравнить с эталонной трассировкой**

## 5 задач трассировок

- Мониторинг распределённых транзакций
- Оптимизация производительности (latency)
- Поиск причин (ошибок, деградации)
- Анализ зависимости сервисов
- Распространение контекста

## ~~5 задач трассировок~~

- Мониторинг распределённых транзакций
- Оптимизация производительности (latency)
- Поиск причин (ошибок, деградации)
- Анализ зависимости сервисов
- Распространение контекста

# Множество задач трассировок

- Мониторинг распределённых транзакций
- Оптимизация производительности (latency)
- Поиск причин (ошибок, деградации)
- Анализ зависимости сервисов
- Распространение контекста

# Множество задач трассировок

- Мониторинг распределённых транзакций
- Оптимизация производительности (latency)
- Поиск причин (ошибок, деградации)
- Анализ зависимости сервисов
- Распространение контекста
- 
- 
-

Q / A



chnl\_GregoryKoshelev



gnkoshelev