Яндекс 360

Отказоустойчивый конечный автомат на Java и SQL

который не дает встрече в Яндекс Телемосте раздвоиться

Дмитрий Некрылов

14 Октября 2023

Об авторе

Дмитрий Некрылов —
ведущий разработчик
Яндекс Телемоста

15 лет опыта коммерческой разработки на java

- *Order Capture & Management*
- *Application Performance Monitoring*
- *Видео-конференц-связь*

Давайте знакомиться!

Antenna, Unsplash.com

О чем этот доклад

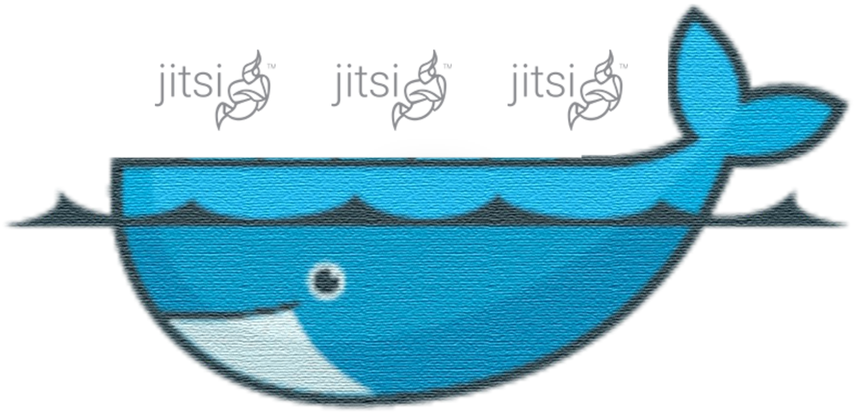
Многопользовательские in-memory сессии

https://www.livemaster.ru/topic/888691-pritcha-grabli

О чем этот доклад

Многопользовательские in-memory сессии

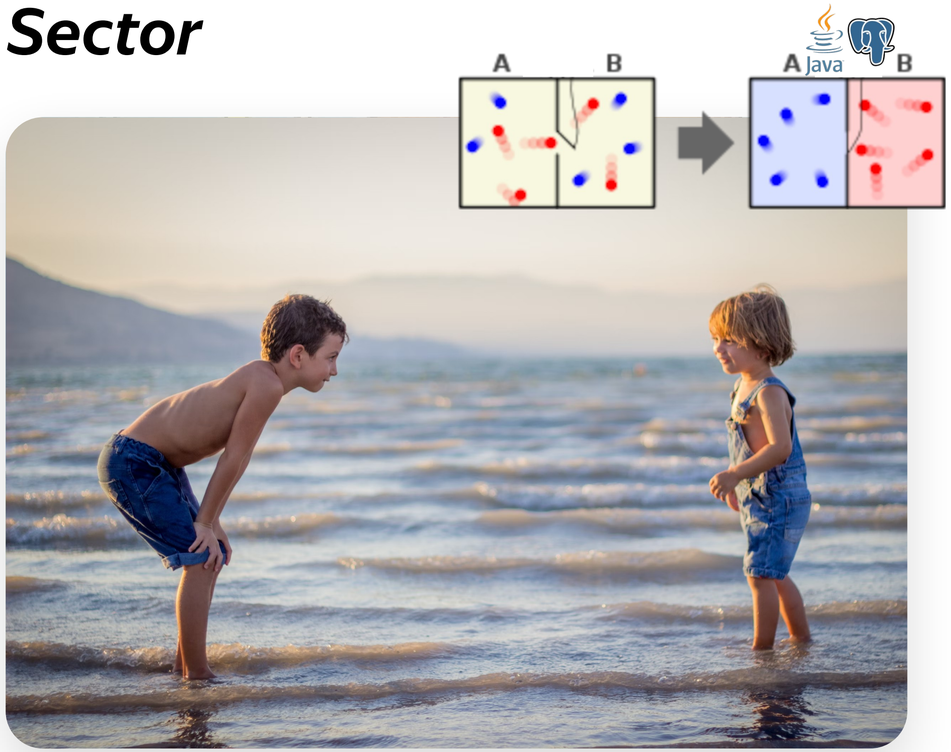
Scaling мульти-кластеров jitsi



https://www.livemaster.ru/topic/888691-pritcha-grabli, https://hub.docker.com/_/docker, https://jitsi.org/

Зачем этот доклад

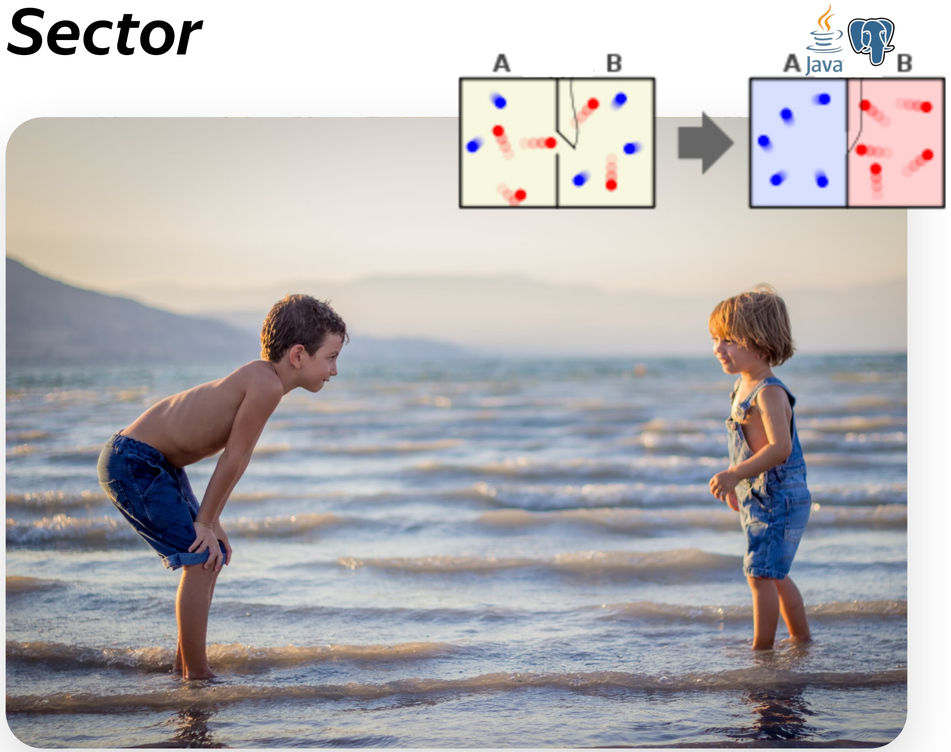
System Performance Sector



Wikipedia, Java (programming language); Wikipedia, Демон Максвелла

Зачем этот доклад

System Performance Sector

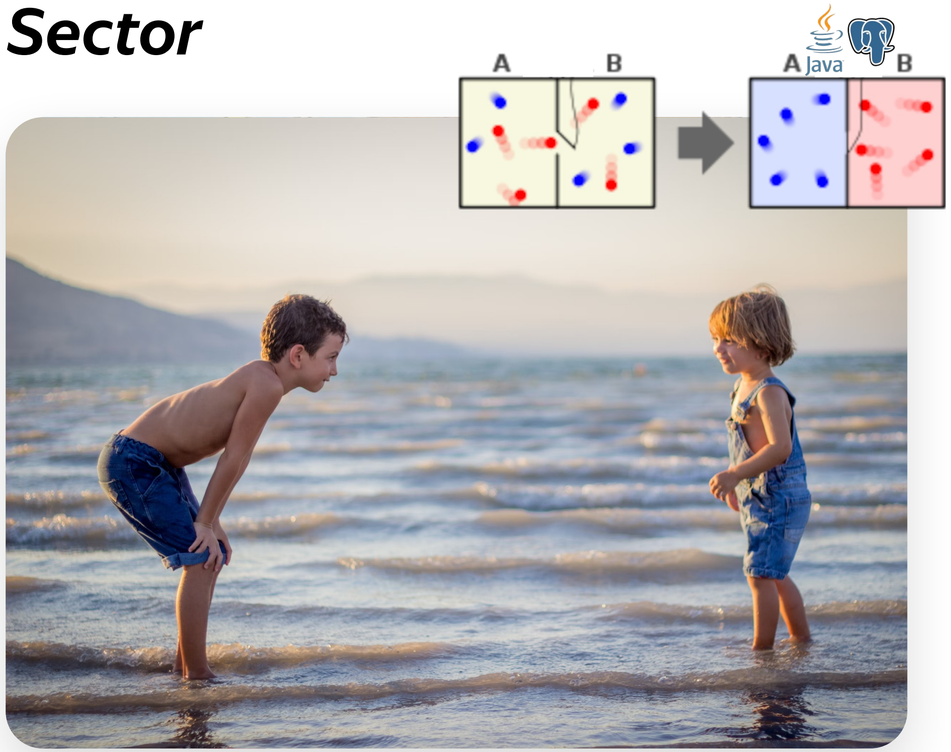


Если нельзя, но очень хочется, то можно!

Wikipedia, Java (programming language); Wikipedia, Демон Максвелла; Wikipedia, PostgreSQL; Meme-Arsenal, «Конечный автомат»; Limor Zellermayer, unsplash.com

Зачем этот доклад

System Performance Sector



Если нельзя, но очень хочется, то можно!

Wikipedia, Java (programming language); Wikipedia, Демон Максвелла; Wikipedia, PostgreSQL; Meme-Arsenal, «Конечный автомат»; Limor Zellermayer, unsplash.com

О чем не пойдет речь

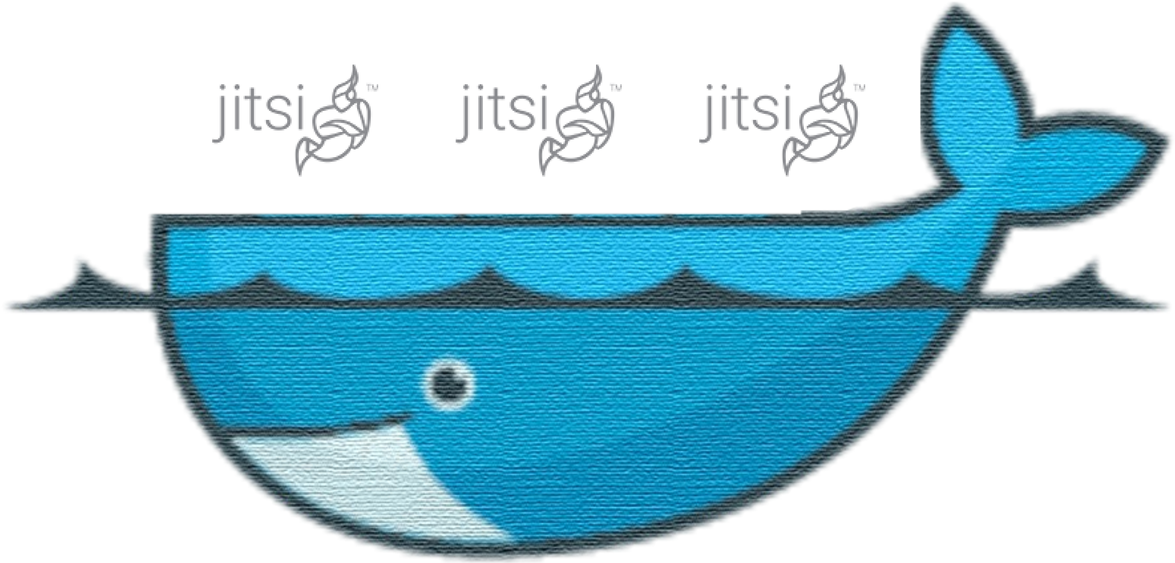
Почему Jitsi

Тюнинг Prosody

О тонкостях WebRTC

Если нельзя, но очень хочется — то можно!

- *Скалировать маленький многопользовательский сервис на весь мир!*



https://hub.docker.com/_/docker; https://jitsi.org/

Если нельзя, но очень хочется — то можно!

- *Менять Primary Key записей в многопоточной среде!*

Tom Roberts, Unsplash.com

Если нельзя, но очень хочется — то можно!

- *Улучшить производительность!*

Yeshi Kangrang, Unsplash.com

На базе Jitsi Meet

- *Open-source*
- *Широко используется*
- *Для небольших офисов*

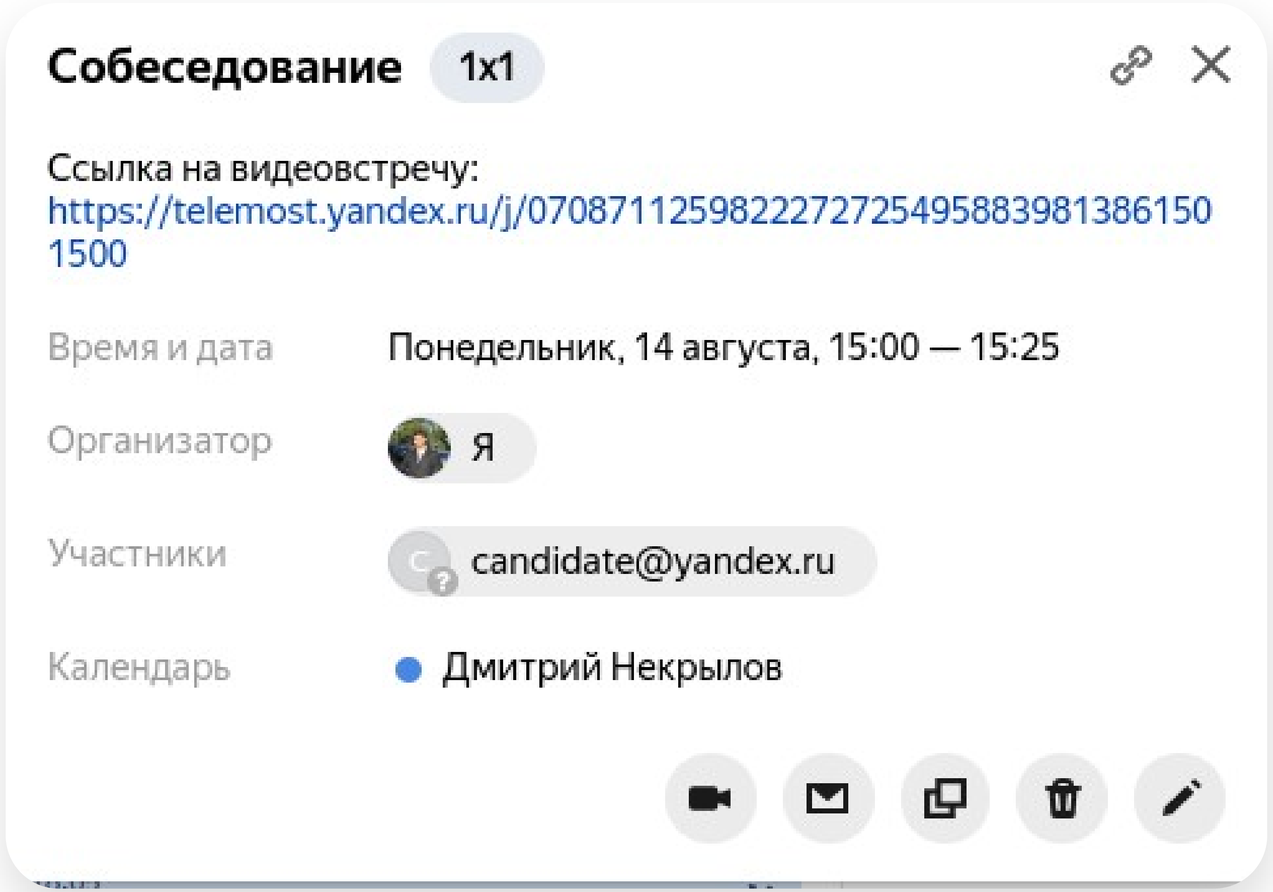
https://jitsi.org/

В облаке

- Высокая нагрузка
- Design for failure
- Smart Endpoints,
  Dumb Pipelines

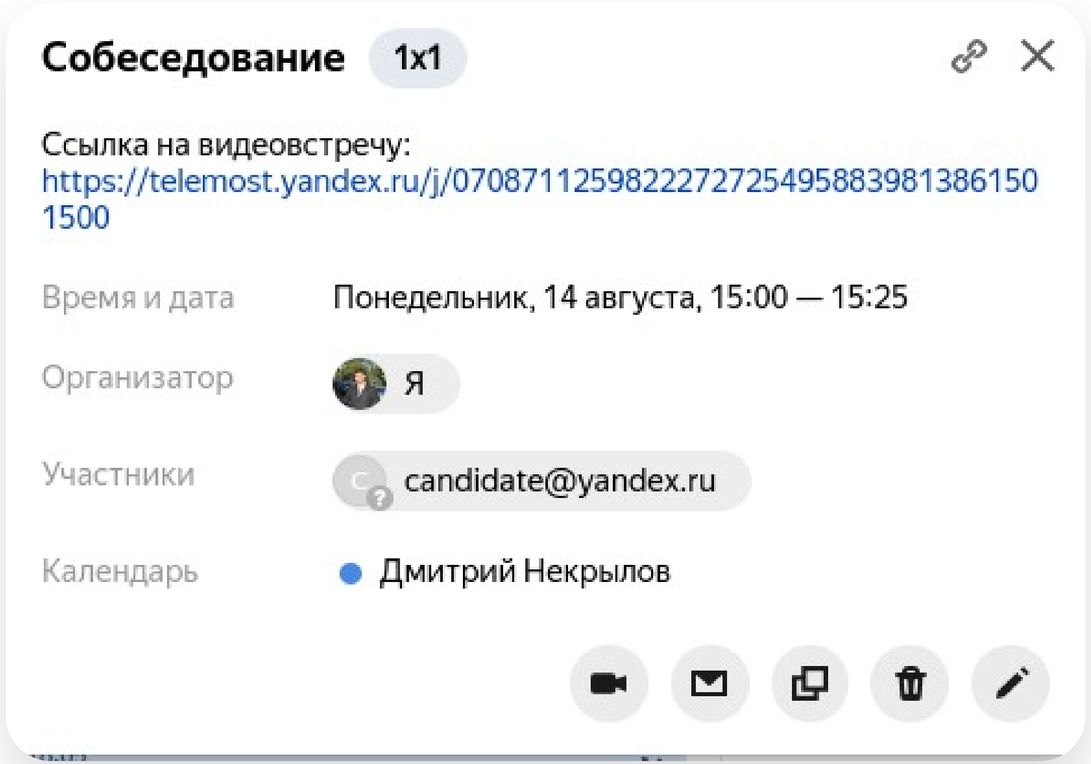
WallpaperFlare.com

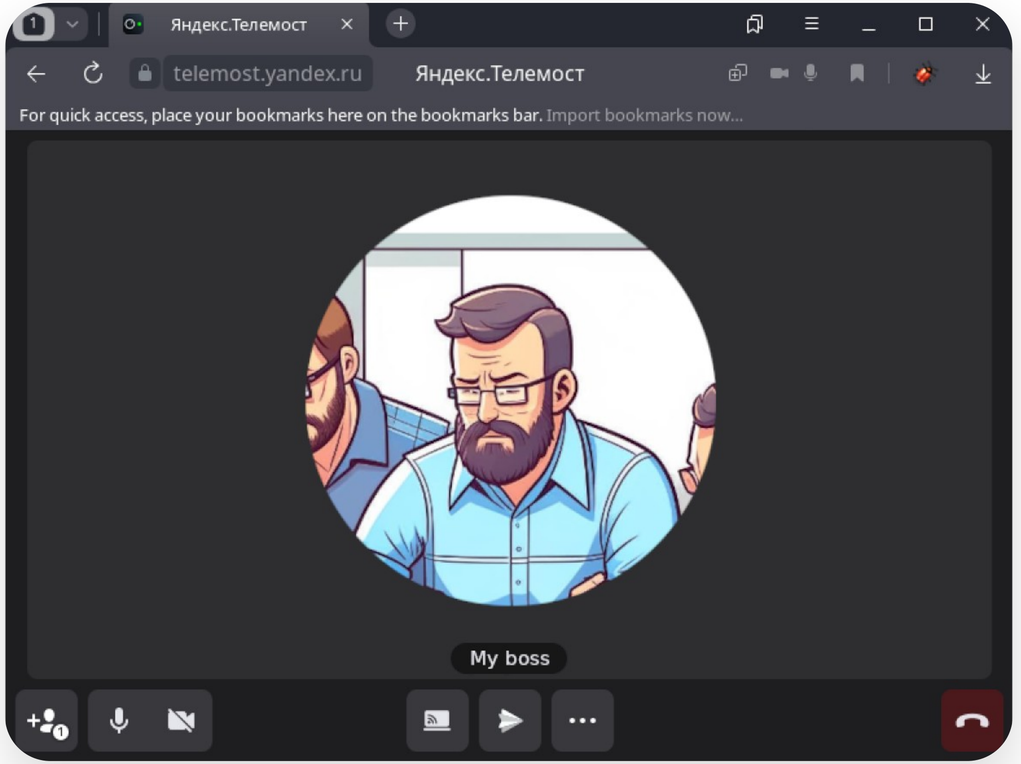
Ссылками делятся заранее



Команда телемоста

Однажды на собеседовании



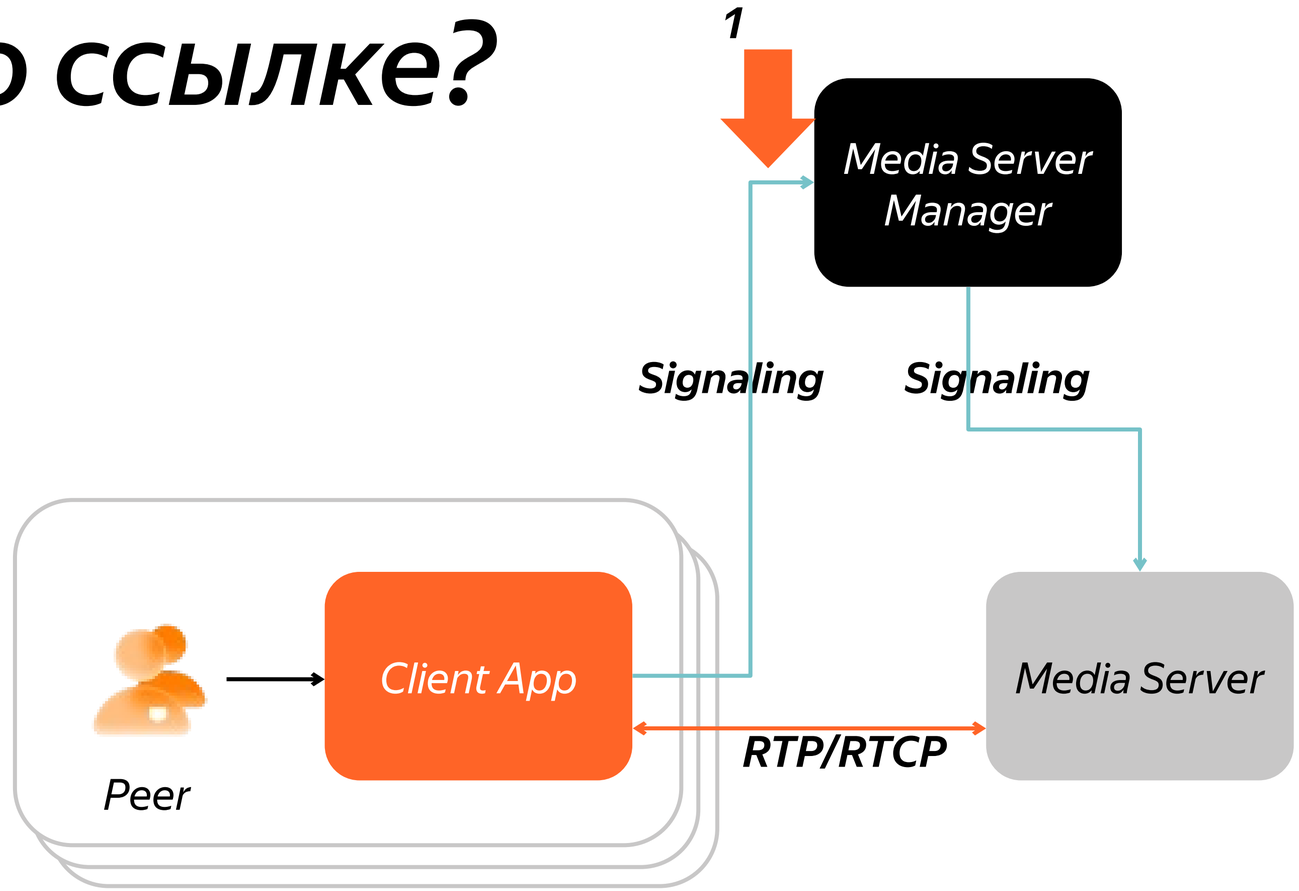


Котаны поняли проблему

Собеседование проводил НАШ начальник

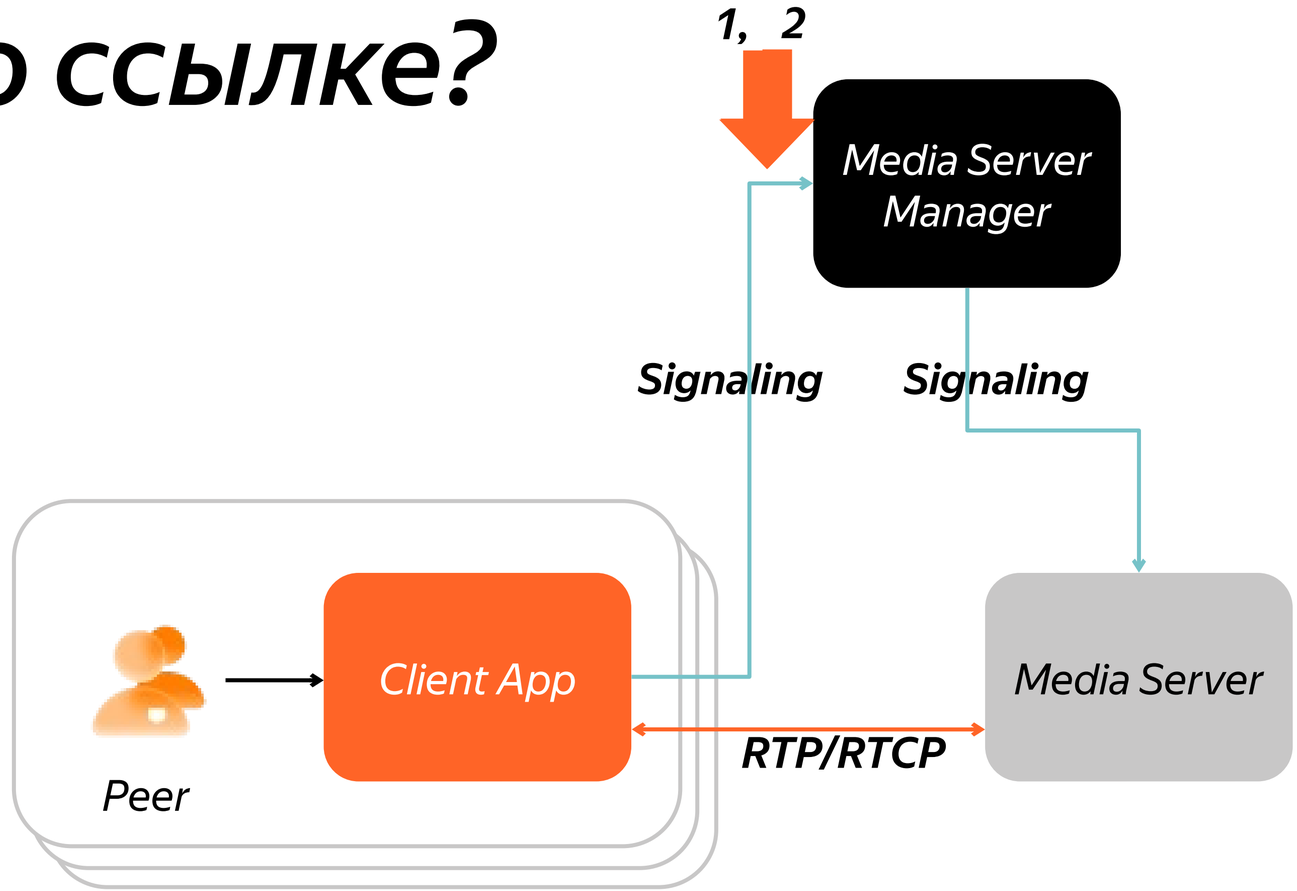
Как можно не встретиться по ссылке?

1. Запросить параметры медиа сервера

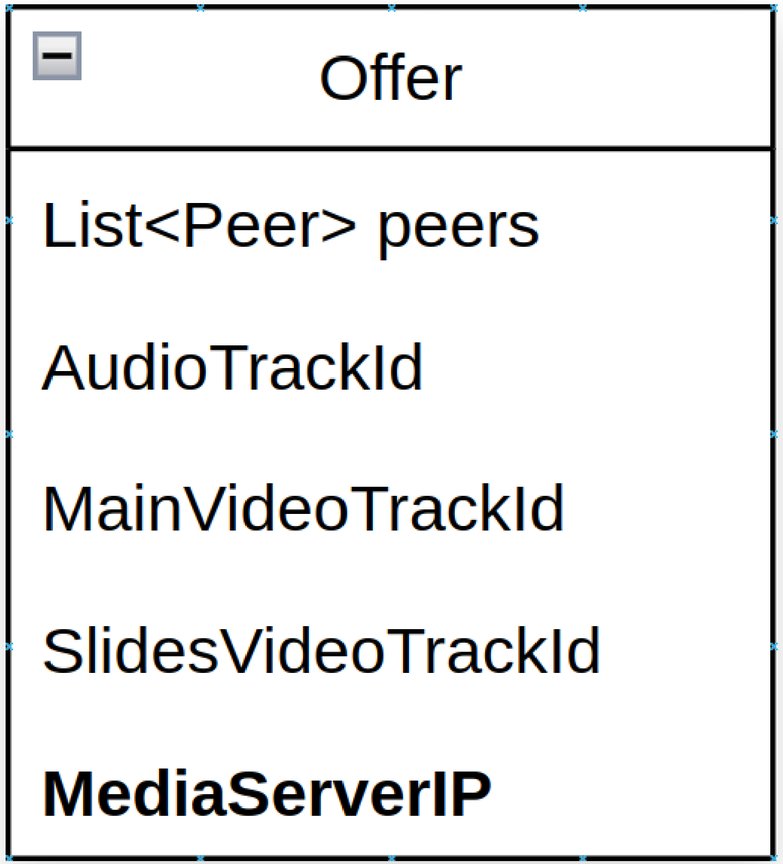


Как можно не встретиться по ссылке?

1. Запросить параметры медиа сервера
2. Дождаться **Offer** — приглашения от медиа сервера

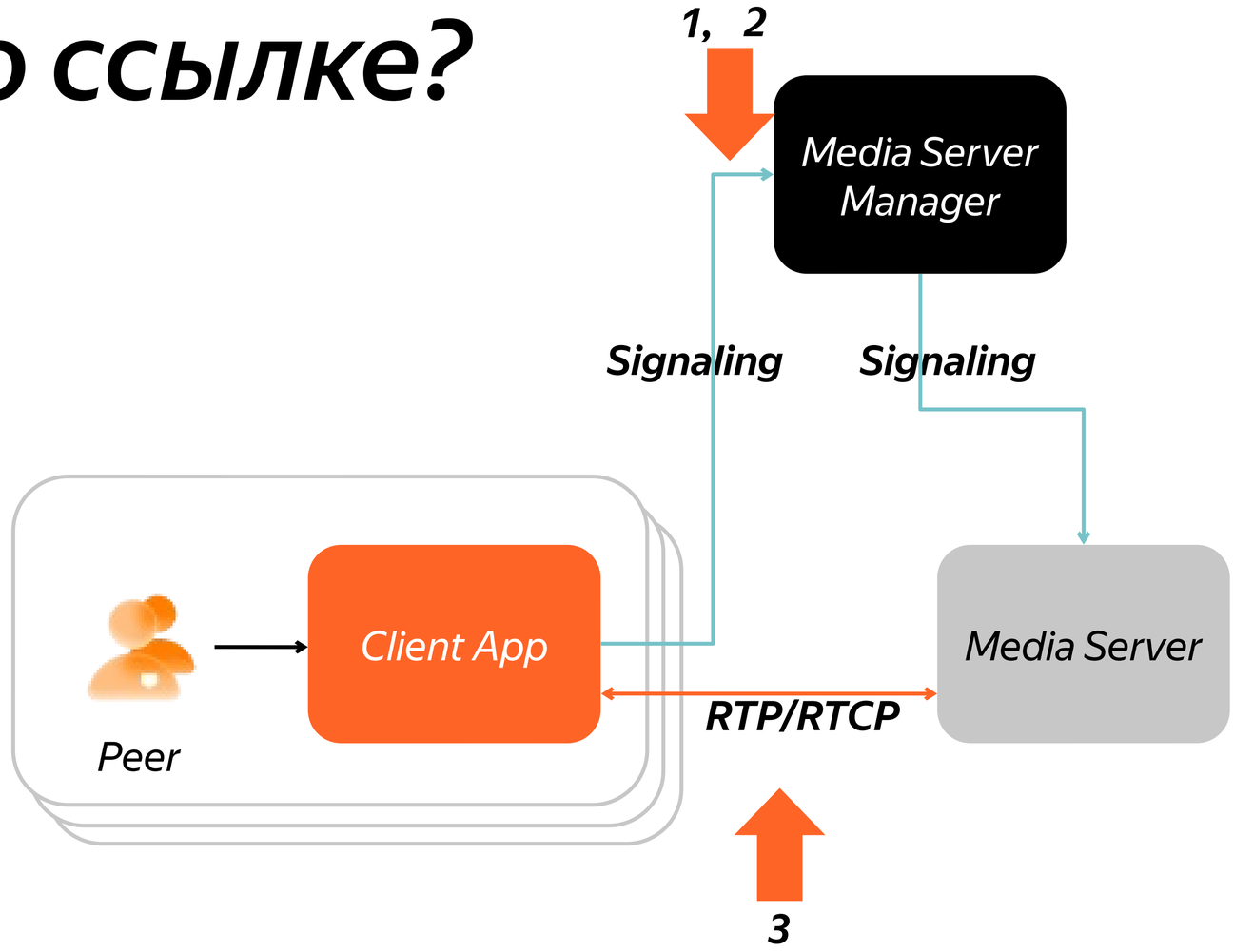


Offer


Offer
List<Peer> peers
AudioTrackId
MainVideoTrackId
SlidesVideoTrackId
MediaServerIP

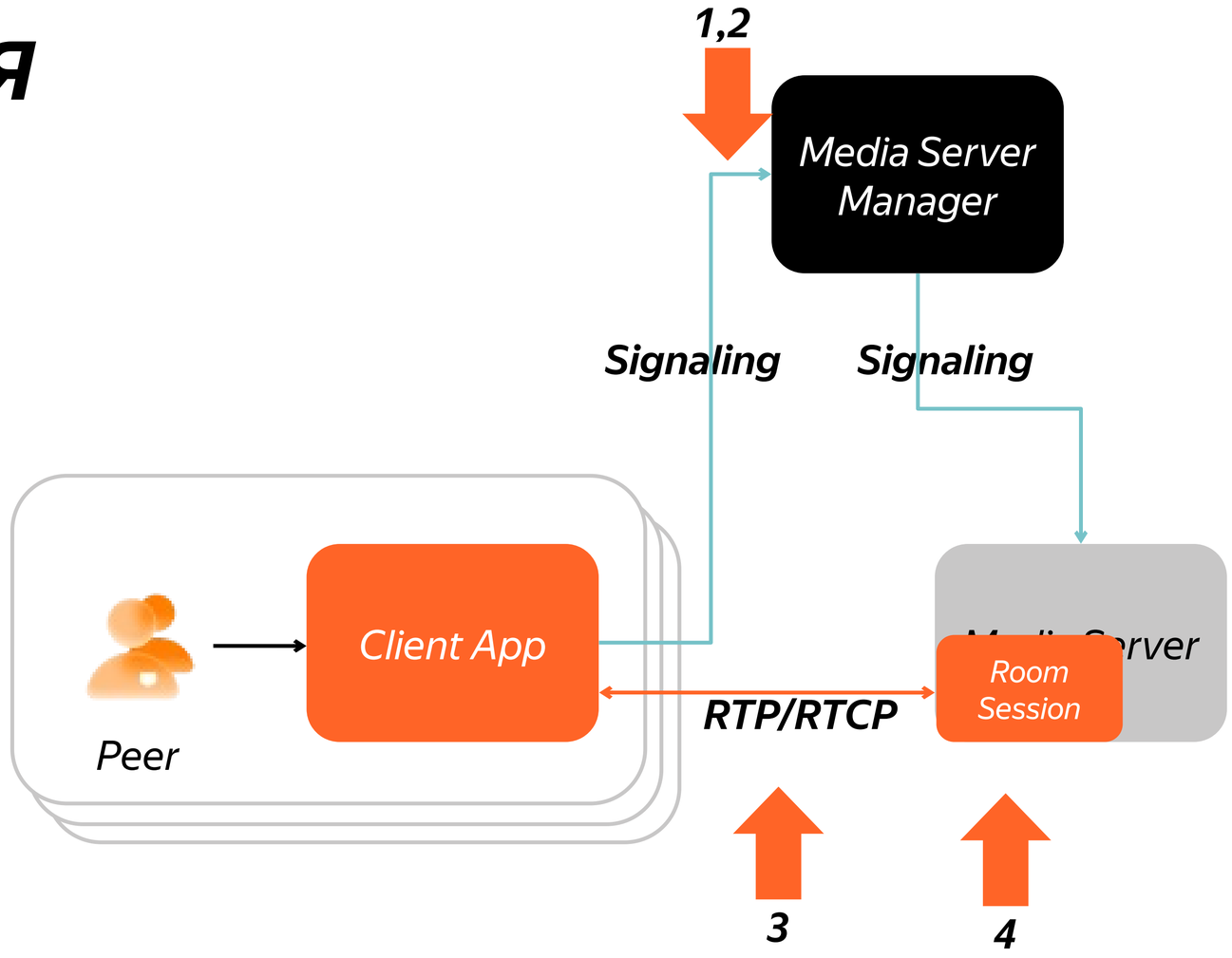
Как можно не встретиться по ссылке?

1. Запросить параметры медиа сервера
2. Дождаться Offer — приглашения от медиа сервера
3. Установить соединение



На медиа сервере всех ждет сессия

1. Запросить параметры медиа сервера
2. Дождаться Offer — приглашения от медиа сервера
3. Установить соединение
4. Подключиться к общей сессии



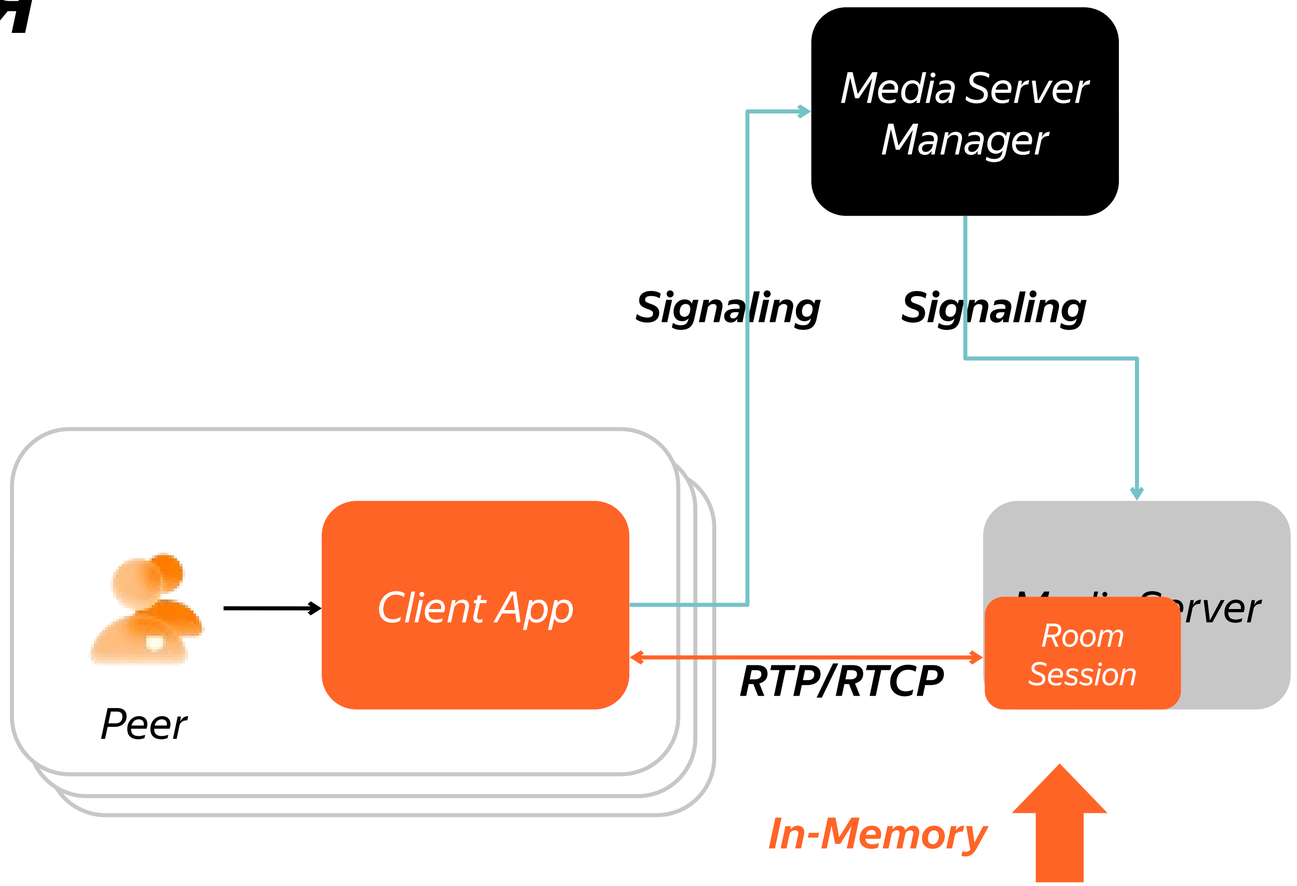
Виртуальная переговорка

Benjamin Child, unsplash.com

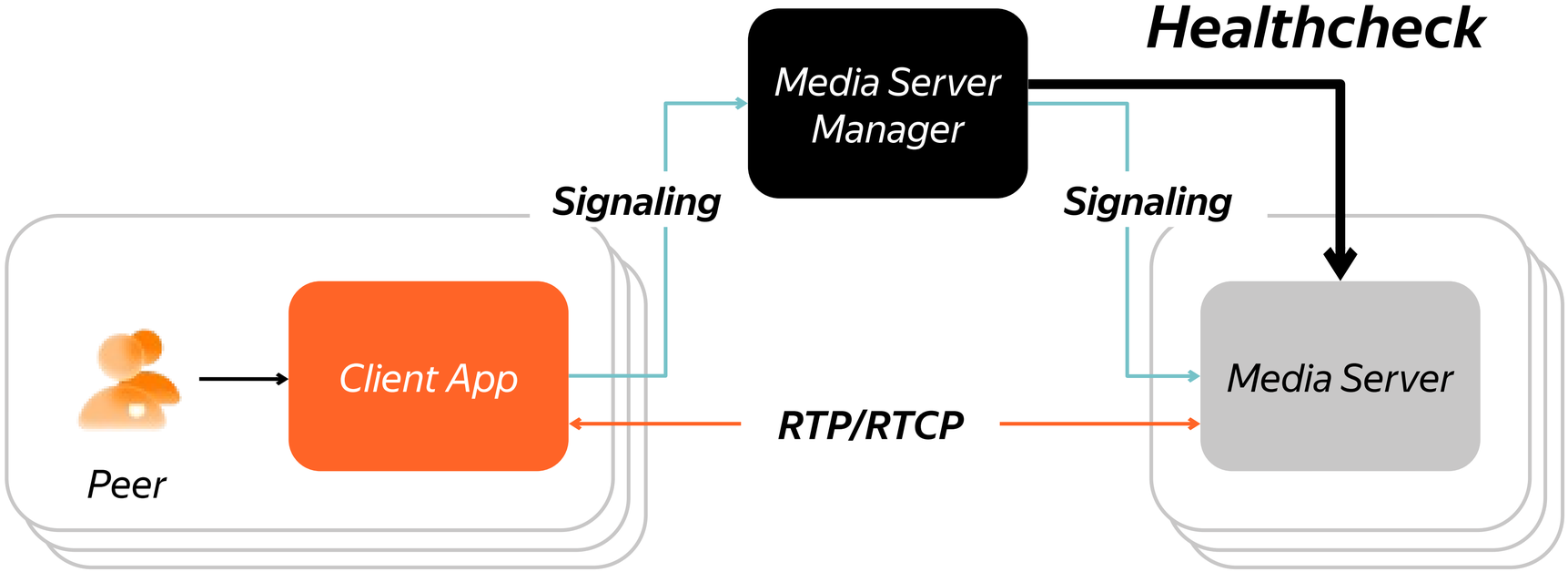
На медиа сервере всех ждет сессия

1. Запросить параметры медиа сервера
2. Дождаться Offer — приглашения от медиа сервера
3. Установить соединение
4. Подключиться к общей сессии

Не дышать на медиа сервер

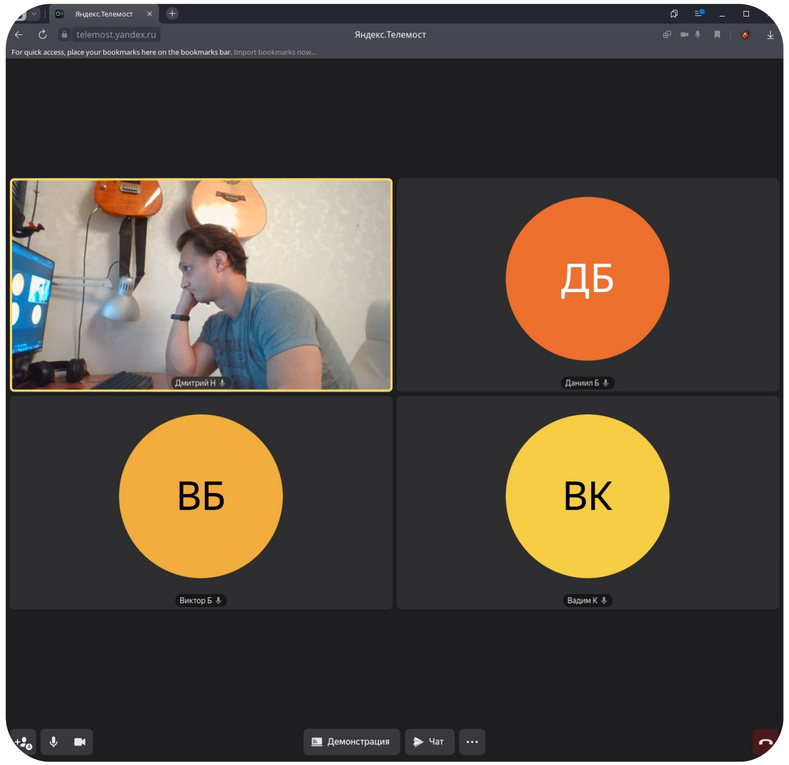


Как можно не встретиться по ссылке: гипотеза

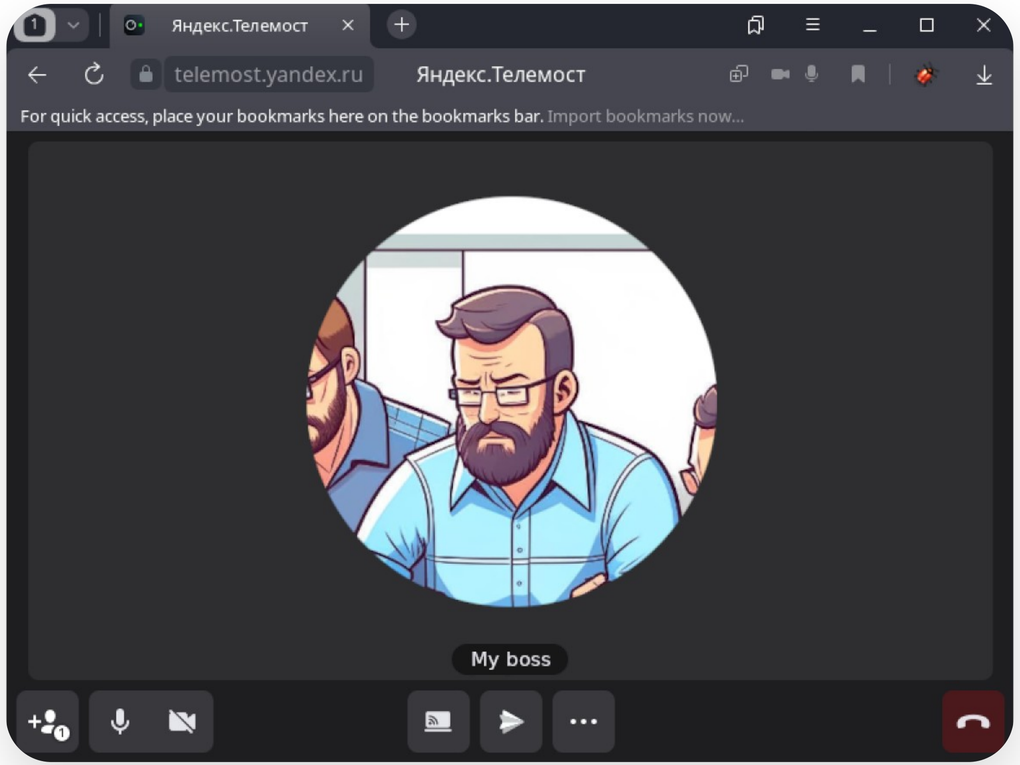


Собираемся в телемосте, проверяем

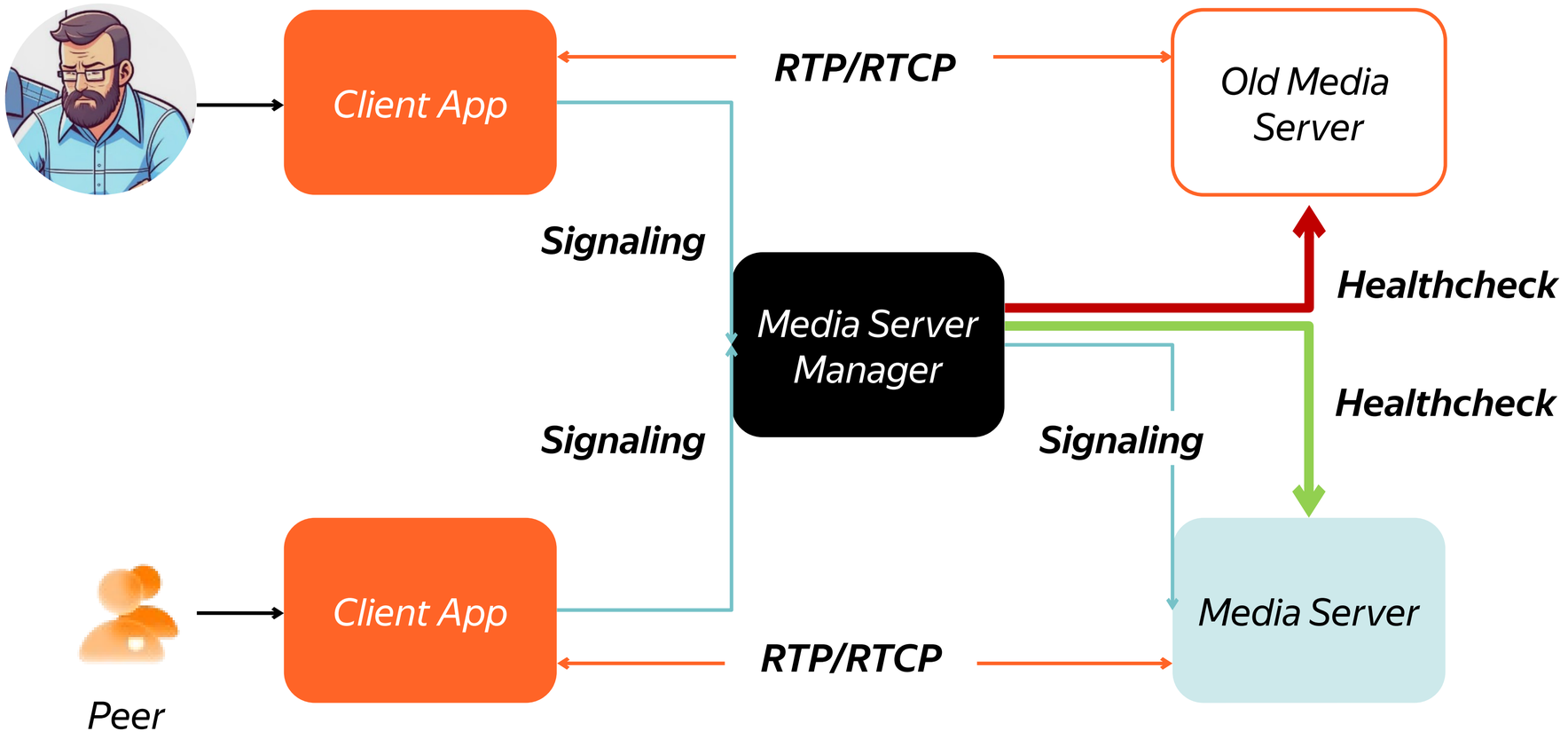
Участники у меня



Участники у начальника

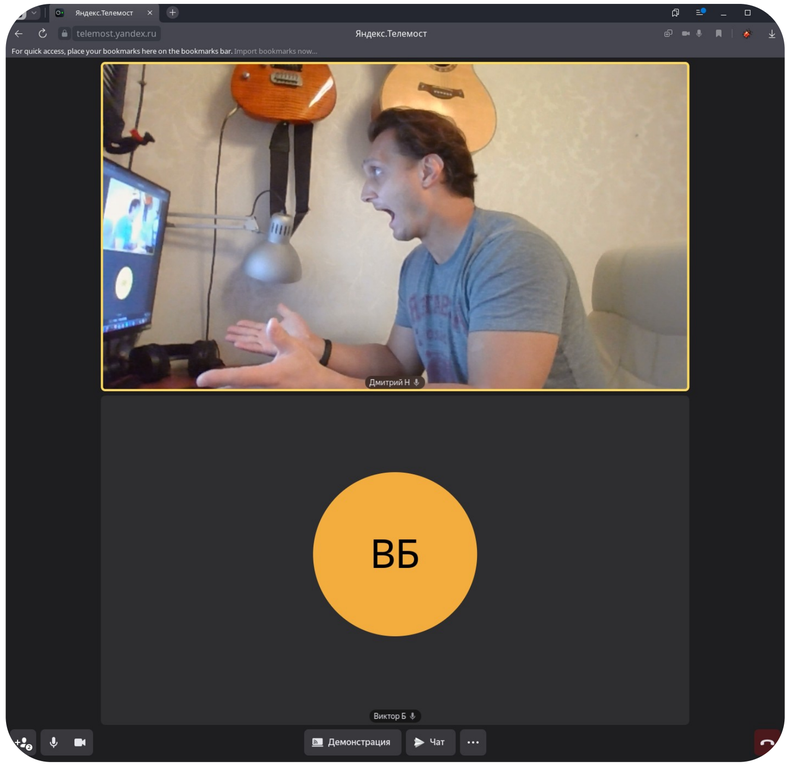


Подтверждаем гипотезу по логам

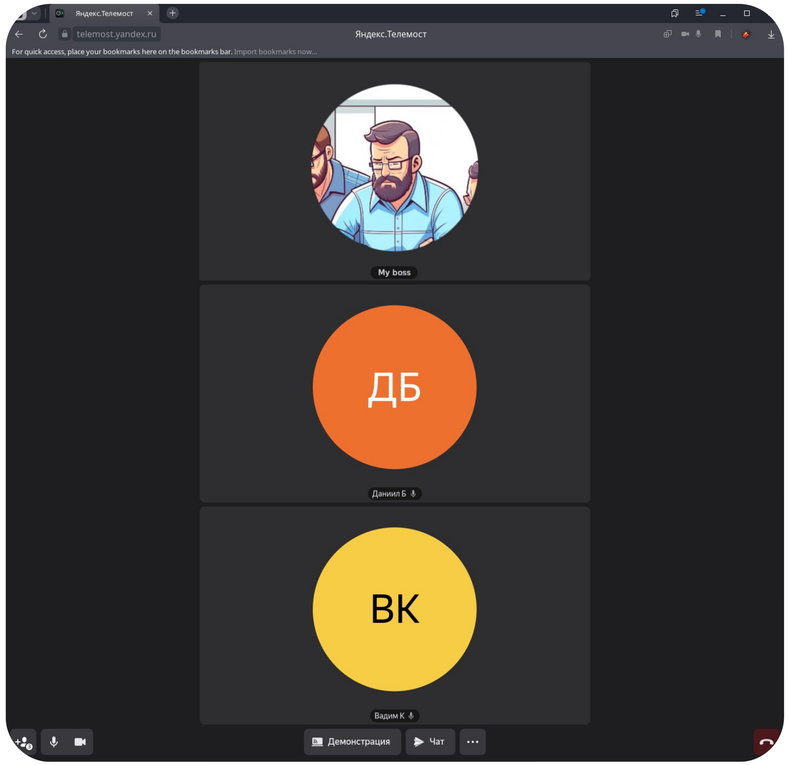


Еще через минуту

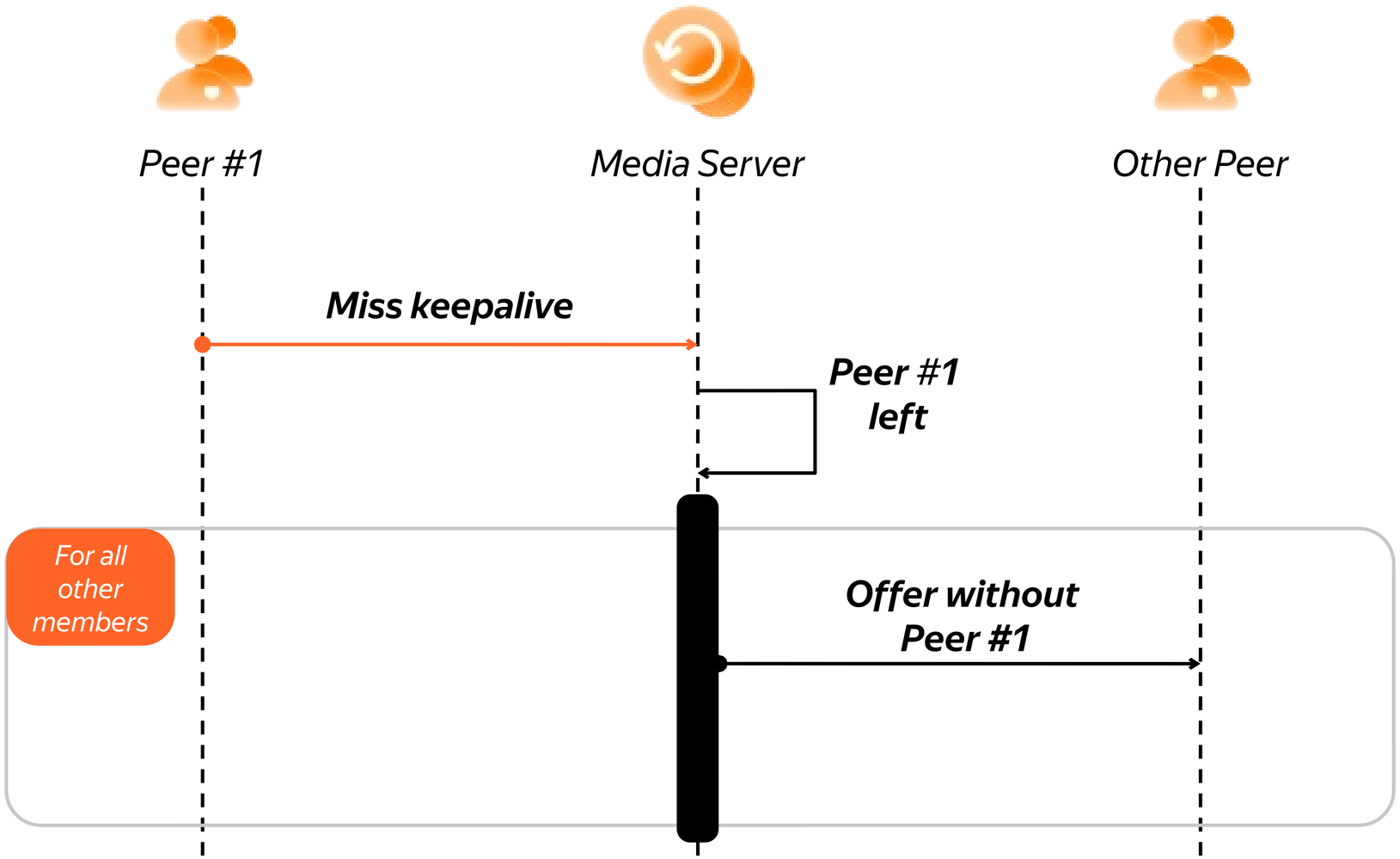
Участники у меня



Участники у начальника



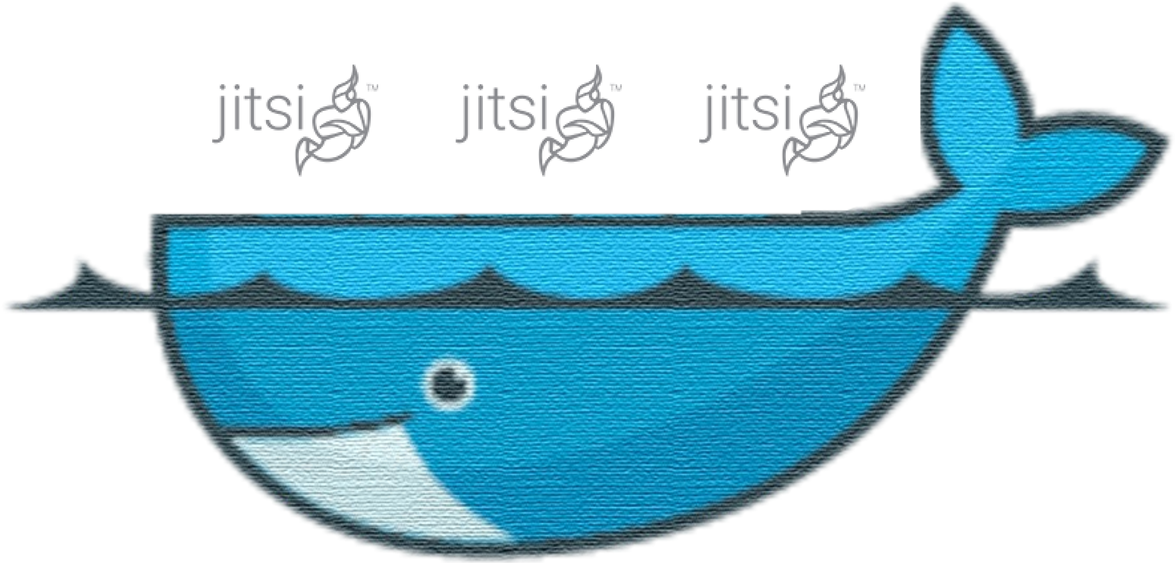
Отключение другого участника



Рут коз понятен

С шаблонами PaaS нужно аккуратнее

- *Health Probes для медиа серверов*
- *Push нотификации с Offer*



https://hub.docker.com/_/docker; https://jitsi.org/

Котаны пошли разбираться

Что дальше? С чего начать?

Нужно это починить

Что дальше? С чего начать?

Нужно это починить

Понять, как такое чинить

Что дальше? С чего начать?

Нужно это починить

Понять, как такое чинить

Сформулировать требования к результату

Требования

Конференция не должна раздваиваться больше, чем на

5 секунд

в 99 квантиле

Что дальше? С чего начать?

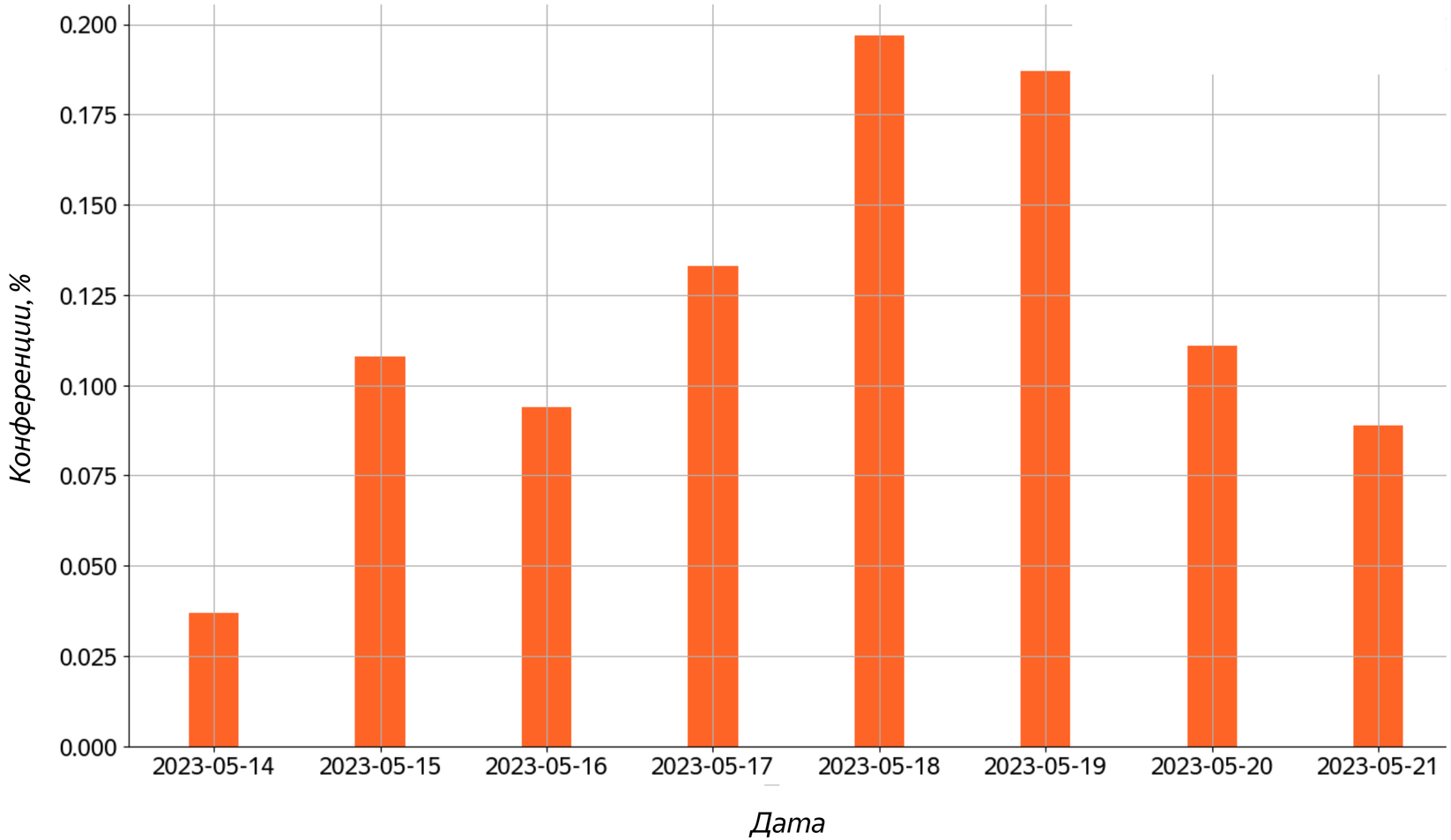
Нужно это починить

Понять, как такое чинить

Сформулировать требования к результату

Убедиться в актуальности проблемы

Масштаб проблемы



Картина решения

- Микросервис
- Внешний наблюдатель
- Не трогаем основу сервиса

Nick Fewings https://unsplash.com/

Решение

Валидировать отчеты Media Серверов о

Присоединении
участников

Решение

Валидировать отчеты Media Серверов о

Присоединении участников

Успешном установлении соединения

Решение

Валидировать отчеты Media Серверов о

Присоединении участников

Успешном установлении соединения

Отсоединении участников

Room Session vs Room Meida Session

Telemost meeting https://telemost.yandex.ru/j/41210079380590

	Peer #1	Peer #2	Peer #3	Peer #4	Peer #5	Peer #6
Room Session №4			Room Media Session			
Room Session №3	Room Media Session					
Room Session №2		Room Media Session				
Room Session №1				Room Media Session	Room Media Session	Room Media Session

Room Session & Room Meida Session

Room Session

Многопользовательская сессия на медиа сервере

Координаты:
***room_session_id** — назначается медиа сервером*

Room Media Session

Сессия пользователя внутри Room Session

Координаты:
room_session_id
***peer_id** — назначается бекендом*

Внешний наблюдатель

- *Внешний по отношению к медиа серверу*

Внешний наблюдатель

- *Внешний по отношению к медиа серверу*
- *Знает, какой медиа сервер актуален*

Внешний наблюдатель

- *Внешний по отношению к медиа серверу*
- *Знает, какой медиа сервер актуален*
- *Знает, какая Room Session актуальна*

Внешний наблюдатель

- Внешний по отношению к медиа серверу
- Знает, какой медиа сервер актуален
- Знает, какая Room Session актуальна
- Зовет в актуальную сессию потеряшек

Внешний наблюдатель

- *Внешний по отношению к медиа серверу*
- *Знает, какой медиа сервер актуален*
- *Знает, какая Room Session актуальна*
- *Зовет в актуальную сессию потеряшек*
- *Знает, какие Room Session устарели*

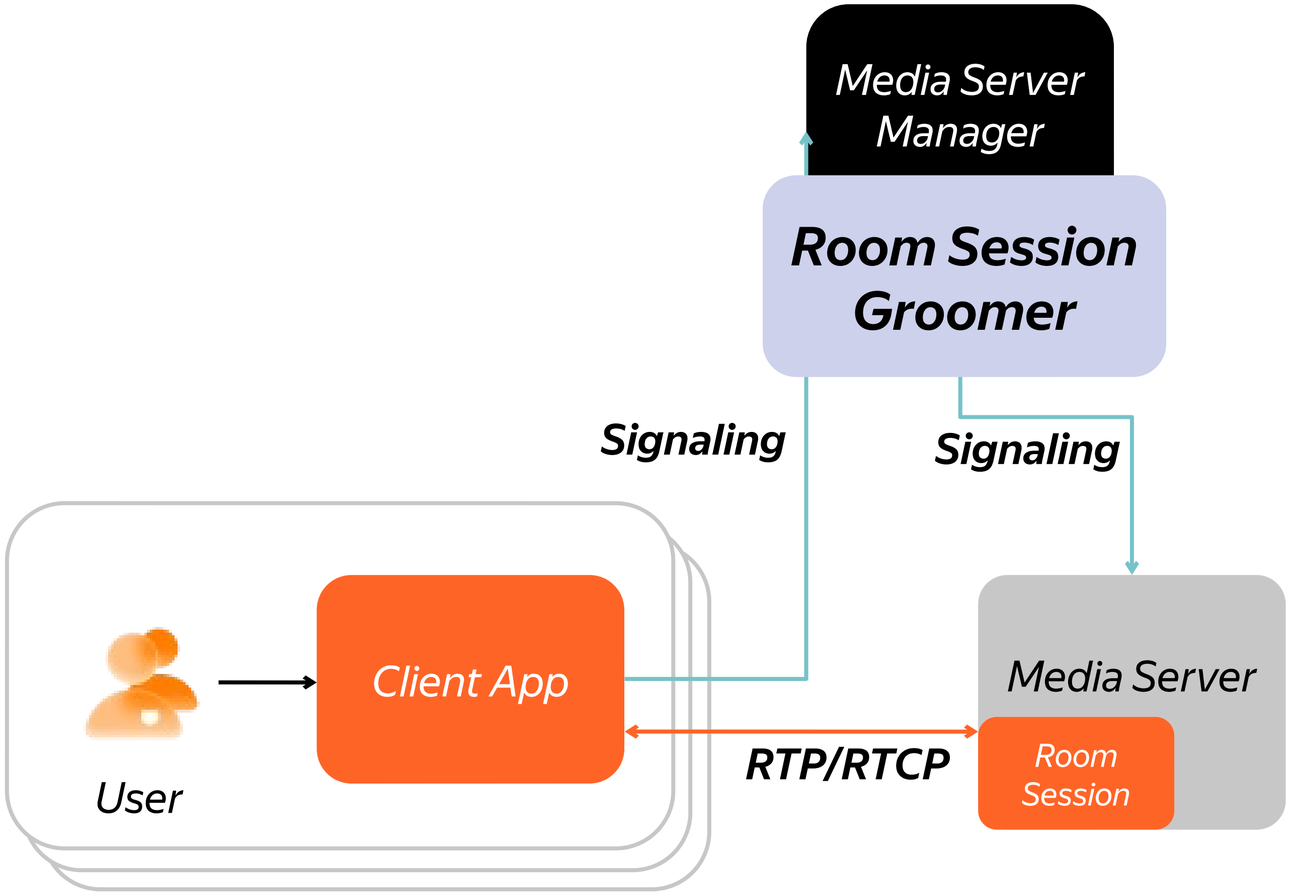
Внешний наблюдатель

- Внешний по отношению к медиа серверу
- Знает, какой медиа сервер актуален
- Знает, какая Room Session актуальна
- Зовет в актуальную сессию потеряшек
- Знает, какие Room Session устарели

Nick Fewings https://unsplash.com/

Room Session Groomer

- Внешний по отношению к медиа серверу
- Знает, какой медиа сервер актуален
- Знает, какая Room Session актуальна
- Зовет в актуальную сессию потеряшек
- Знает, какие Room Session устарели

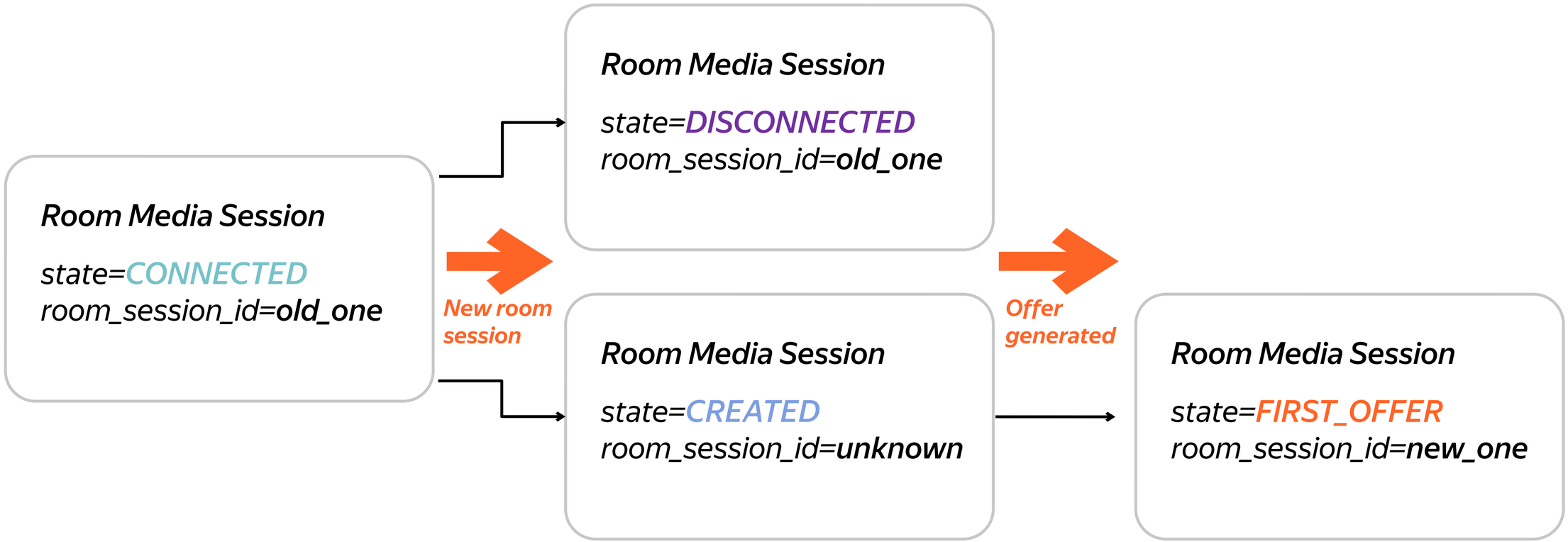


Что делать с нелегитимными офферами?

- *Любую неизвестную **Room Session** признаем **новой***

- *Остальные **Room Session***
  - *храним в истории*
  - *признаем **нелегитимными***
  - *не принимаем в них офферы*

Что делать с потеряшками?



Отчеты медиа сервера

Сообщение	Параметры	Описание
First Offer	• peer_id • room_session_id	Кому и в какую **Room Session** был отослан оффер Бекенд может принять решение не передавать оффер клиенту
Connected	• peer_id • room_session_id	Кто и в какой сессии подключился
Disconnected	• peer_id • room_session_id	Медиа Сервер больше не пустит пользователя в эту **Room Session**

Состояния Room Media Session

Состояние	Координаты	Описание
Created	• peer_id • rms_unknown	Пользователь изъявил желание подключиться
First Offer	• peer_id • room_session_id	Медиа сервер отправил оффер пользователю При необходимости сформировал новую **Room Session**
Connected	• peer_id • room_session_id	Медиа сервер подтвердил, что пользователь подключился
Disconnected	• peer_id • room_session_id	• Пользователь вышел из конференции • Пользователь «въехал в туннель», и прошла минута • **Room Session** признана нелегитимным, пользователю отправлен сигнал на переподключение

Меняется Primary Key записи

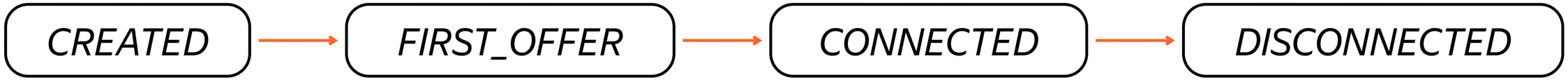
Состояние	Координаты	Описание
Created	• *peer_id* • ***rms_unknown***	*Пользователь изъявил желание подключиться*
First Offer	• *peer_id* • ***room_session_id***	*Медиа сервер отправил оффер пользователю* *При необходимости сформировал новую* ***Room Session***
Connected	• *peer_id* • *room_session_id*	*Медиа сервер подтвердил, что пользователь подключился*
Disconnected	• *peer_id* • *room_session_id*	• *Пользователь вышел из конференции* • *Пользователь «въехал в туннель», и прошла минута* • ***Room Session*** *признана нелегитимным, пользователю отправлен сигнал на переподключение*

Движение только вперед

Состояние	Координаты	Описание
Created	• peer_id	Пользователь изъявил желание подключиться
First Offer	• peer_id • room_session_id	Медиа сервер отправил оффер пользователю При необходимости сформировал новую **Room Session**
Connected	• peer_id • room_session_id	Медиа сервер подтвердил, что пользователь подключился
Disconnected	• peer_id • room_session_id	• Пользователь вышел из конференции • Пользователь «въехал в туннель», и прошла минута • **Room Session** признана нелегитимным, пользователю отправлен сигнал на переподключение

По команде медиасервера

Найти в базе запись одного из предыдущих состояний, и перевести ее в следующее



Команда всегда с параметрами (peer_id, room_session_id)

По команде медиасервера

Если не было, нужно создать

CREATED → ... меняет PK

(peer_id, <rms_unknown>) → (peer_id, room_session_id)

нельзя раздвоить

По команде медиасервера

01 Update записи по (peer_id, rs_id)

(если не нашлось)

02 Update Created записи (peer_id, RMS_UNKNOWN) (peer_id, rs_id)

(если тоже не успешен)

03 Insert записи (peer_id, rs_id)

По команде медиасервера

01 Update записи по (peer_id, rs_id)

неуспешен если такой не нашлось

(если не нашлось)

02 Update Created записи (peer_id, RMS_UNKNOWN) (peer_id, rs_id)

неуспешен если такой не нашлось либо on conflict

(если тоже не успешен)

03 Insert записи (peer_id, rs_id)

неуспешен on conflict

Такой дизайн

- Вводим Room Session
- Вводим Room Media Session
- Следим, что все совещаются в одной Room Session

Unsplash.com, Touann Gatouillat Vergos

Котаны пошли делать MVP

Структура БД

```
create table room_media_sessions
(
    room_id            uuid                        not null,
    peer_id            uuid                        not null,
    room_session_id    uuid                        not null,
    created_at         timestamptz(3)              not null,
    first_offer_at     timestamptz(3),
    connected_at       timestamptz(3),
    disconnected_at    timestamptz(3),
    state              room_media_session_state not null,
);
```

Структура БД

```
CONSTRAINT check_room_media_sessions_state_requirements CHECK (
            (state = 'CREATED'                  and room_session_id =  'rms_unknown') or
            (state = 'FIRST_OFFER_RECEIVED'   and room_session_id <> 'rms_unknown') or
            (state = 'CONNECTED'                and room_session_id <> 'rms_unknown') or
            (state = 'DISCONNECTED'           and room_session_id <> 'rms_unknown') or
            state = 'ARCHIVED'
    )
```

Структура БД

```
CONSTRAINT check_room_media_sessions_state_requirements CHECK (
            (state = 'CREATED'                 and room_session_id =  'rms_unknown') or
            (state = 'FIRST_OFFER_RECEIVED'    and room_session_id <> 'rms_unknown') or
            (state = 'CONNECTED'               and room_session_id <> 'rms_unknown') or
            (state = 'DISCONNECTED'            and room_session_id <> 'rms_unknown') or
            state = 'ARCHIVED'
    )
```

Структура БД

```
CONSTRAINT check_room_media_sessions_state_requirements CHECK (
            (state = 'CREATED'                 and room_session_id =  'rms_unknown') or
            (state = 'FIRST_OFFER_RECEIVED'    and room_session_id <> 'rms_unknown') or
            (state = 'CONNECTED'               and room_session_id <> 'rms_unknown') or
            (state = 'DISCONNECTED'            and room_session_id <> 'rms_unknown') or
            state = 'ARCHIVED'
    )
```

Структура БД

```
CONSTRAINT check_room_media_sessions_state_requirements CHECK (
            (state = 'CREATED'                 and room_session_id =  'rms_unknown') or
            (state = 'FIRST_OFFER_RECEIVED'    and room_session_id <> 'rms_unknown') or
            (state = 'CONNECTED'               and room_session_id <> 'rms_unknown') or
            (state = 'DISCONNECTED'            and room_session_id <> 'rms_unknown') or
            state = 'ARCHIVED'
    )
```

Структура БД

```
CONSTRAINT check_room_media_sessions_state_requirements CHECK (
            (state = 'CREATED'                 and room_session_id =  'rms_unknown') or
            (state = 'FIRST_OFFER_RECEIVED'    and room_session_id <> 'rms_unknown') or
            (state = 'CONNECTED'               and room_session_id <> 'rms_unknown') or
            (state = 'DISCONNECTED'            and room_session_id <> 'rms_unknown') or
            state = 'ARCHIVED'
    )
```

Управляющая команда

```
@Data
@Builder
public static class UpsertRMSRequest {
    @Nonnull
    String roomId;
    @Nonnull
    String roomSessionId;
    @Nonnull
    String peerId;
    //CREATED, FIRST_OFFER_RECEIVED, CONNECTED, DISCONNECTED, ARCHIVED
    List<RoomMediaSessionState> updatedStates;

    RoomMediaSessionState newState;
}
```

Меняем только предыдущие состояния

```
@Language("SQL")
public static final String UPDATE_BASE_SQL = """
        update room_media_sessions
            set state = case when state in (:updated_states)
                    then :state else state end,
                room_session_id = :room_session_id
        where peer_id = :peer_id
        """;
```

CREATED vs >=FIRST_OFFER

```
@Language("SQL")
public static final String UPDATE_BY_ROOM_SESSION_ID = UPDATE_BASE_SQL + """
            /*-RoomMediaSessionPgDaoImpl-UPDATE_BY_ROOM_SESSION_ID-*/
            and room_session_id = :room_session_id
        """;
@Language("SQL")
public static final String UPDATE_CREATED_ROOM_SESSION = UPDATE_BASE_SQL +
String.format("""
            /*-RoomMediaSessionPgDaoImpl-UPDATE_CREATED_ROOM_SESSION-*/
            and state = 'CREATED'
            and room_session_id = '%s'
        """, UNKNOWN_ROOM_SESSION_ID);
```

Postgres Work Analyzer

Details for all queries

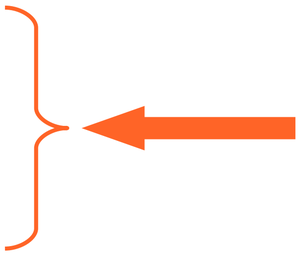
Export CSV

	Execution			I/O Time		Blocks				Temp blocks	
Query	#	Time	Avg time	Read	Write	Read	Hit	Dirtied	Written	Read	Written
SELECT pg_sleep($1)	88.00	9 min 48 s	6 s 687 ms	0	0	0 B	0 B	0 B	0 B	0 B	0 B
SELECT count(*) FROM commandes cmd JOIN lignes_commandes lc ON lc.nume…	16.00	2 min 19 s	8 s 709 ms	6 min 52 s	0	2.05 G	479.01 M	0 B	0 B	0 B	0 B
SELECT COUNT(*) FROM pieces_fournisseurs WHERE cout_piece >= $1	10.00	49 s 642 ms	4 s 964 ms	2 min 27 s	0	457.24 M	75.49 M	0 B	0 B	0 B	0 B
SELECT numero_commande, etat_commande FROM commandes WHERE client_id =…	16.00	40 s 960 ms	2 s 560 ms	1 min 37 s	0	214.58 M	388.17 M	0 B	0 B	0 B	0 B
SELECT * FROM clients cl JOIN contacts co ON co.contact_id = cl.contac…	16.00	29 s 461 ms	1 s 841 ms	0	0	0 B	671.41 M	0 B	0 B	540.63 M	540.63 M
SELECT co.nom FROM clients cl JOIN contacts co ON co.contact_id = cl.c…	16.00	17 s 796 ms	1 s 112 ms	0	0	0 B	680.94 M	0 B	0 B	0 B	0 B
SELECT COUNT(*) FROM pays p JOIN contacts con ON con.code_pays = p.cod…	16.00	10 s 702 ms	668 ms 880 µs	0	0	0 B	680.65 M	0 B	0 B	421.55 M	425.99 M
ALTER TABLE ONLY public.lignes_commandes ADD CONSTRAINT lignes_command…	1.00	4 s 587 ms	4 s 587 ms	523 ms 195 µs	16 ms 971 µs	172.57 M	41.25 G	40.00 K	10.84 M	0 B	0 B
COPY public.lignes_commandes (numero_commande, piece_id, fournisseur_i…	1.00	4 s 528 ms	4 s 528 ms	0	130 ms 242 µs	150.80 M	24.00 K	150.80 M	134.80 M	0 B	0 B
SELECT * FROM contacts	16.00	3 s 188 ms	199 ms 288 µs	0	0	0 B	395.50 M	0 B	0 B	0 B	0 B
SELECT COUNT(*) FROM commandes WHERE date_commande BETWEEN ($1 || $2):…	16.00	2 s 181 ms	136 ms 315 µs	656 ms 442 µs	0	184.45 M	418.30 M	0 B	0 B	0 B	0 B
SELECT COUNT(*) FROM pays p JOIN contacts con ON con.code_pays = p.cod…	16.00	2 s 144 ms	134 ms 16 µs	0	0	0 B	680.04 M	0 B	0 B	0 B	0 B
SELECT con.nom || $1 || code_pays || $2 FROM clients cli JOIN contacts…	16.00	1 s 378 ms	86 ms 152 µs	0	0	0 B	680.94 M	0 B	0 B	0 B	0 B
SELECT nom FROM contacts c JOIN pays p ON p.code_pays = c.code_pays WH…	16.00	1 s 235 ms	77 ms 210 µs	0	0	0 B	412.13 M	0 B	0 B	0 B	0 B
COPY public.pieces (piece_id, nom, fabriquant, marque, type_piece, tai…	1.00	1 s 225 ms	1 s 225 ms	0	39 ms 518 µs	61.70 M	0 B	61.70 M	45.70 M	0 B	0 B
COPY public.pieces_fournisseurs (piece_id, fournisseur_id, quantite_di…	1.00	1 s 34 ms	1 s 34 ms	0	33 ms 2 µs	53.27 M	48.00 K	53.27 M	37.27 M	0 B	0 B
SELECT COUNT(*) FROM commandes WHERE date_commande BETWEEN ($1 || $2):…	16.00	903 ms 88 µs	56 ms 443 µs	122 ms 540 µs	0	157.42 M	445.33 M	0 B	0 B	0 B	0 B
SELECT COUNT(*) FROM pieces_fournisseurs WHERE quantite_disponible < $…	10.00	899 ms 725 µs	89 ms 972 µs	467 ms 187 µs	0	449.74 M	82.99 M	0 B	0 B	0 B	0 B

PGConf.EU 2019, Performance analysis at full power

Как теперь искать потеряшек?

```
@Override
public Collection<String> findOtherActiveRoomSessions(@Nonnull String roomId,
@Nonnull String roomSessionId) {
    return jdbcTemplate.queryForList("""
                        select distinct room_session_id
                        from room_media_sessions
                        where room_id = ?
                          and state in ('FIRST_OFFER_RECEIVED', 'CONNECTED')
                          and disconnected_at is null
                          and room_session_id <> ?
                        """,
            String.class,
            roomId,
            roomSessionId
    );
}
```



Производительность?

Limor Zellermayer, unsplash.com

Используем Partial Index

Postgres Partial Index

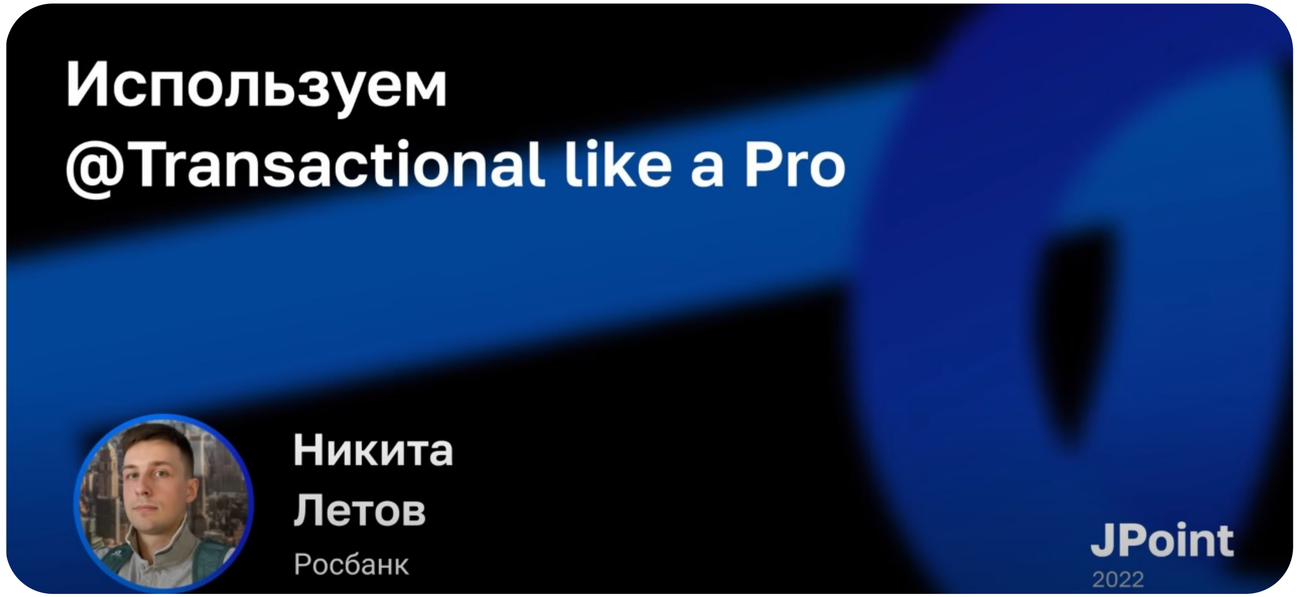
```
CREATE INDEX if not exists idx_room_media_sessions_active_sessions
    ON room_media_sessions (room_id, room_session_id)
    where disconnected_at is null
        and state in ('CREATED', 'FIRST_OFFER_RECEIVED', 'CONNECTED')
```

Нужно сделать update строчки в базе

```
create table room_media_sessions
(
    room_id           uuid                        not null,
    peer_id           uuid                        not null,
    room_session_id   uuid                        not null,
    created_at        timestamptz(3)              not null,
    first_offer_at    timestamptz(3),
    connected_at      timestamptz(3),
    disconnected_at   timestamptz(3),
    state             room_media_session_state    not null,
);
```

С локом в базе

Используем @Transactional like a Pro



С локом в базе: реализация

```
transactionTemplate.executeWithoutResult((action) → {
    roomMediaSessionDao.lockRoomId(upsertRMSRequest.getRoomId());

    int updated;
    updated = roomMediaSessionDao.updateByRoomSessionId(upsertRMSRequest);
    if (updated == 0) {
        updated = roomMediaSessionDao.updateCreatedRoomSession(upsertRMSRequest);
        if (updated == 0) {
            updated = roomMediaSessionDao.insertOrDoNothing(upsertRMSRequest);
        }
    }

    if (updated == 0) {
        throw new RuntimeException(String.format(
                "Failed to update info on room media sessions. 0 rows updated.
request: %s",
                upsertRMSRequest
        ));
    }
});
```

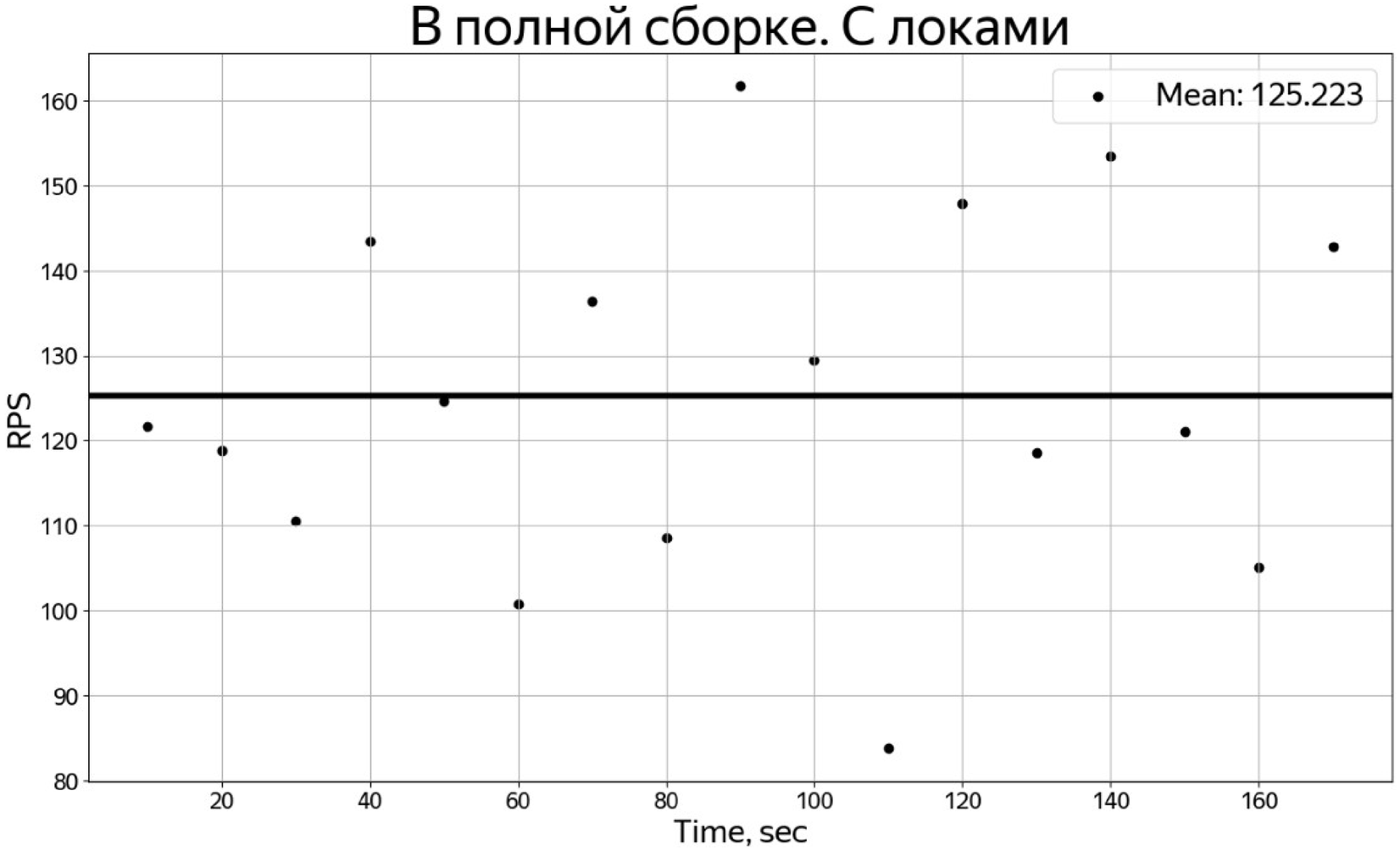
С локом в базе: реализация

```
transactionTemplate.executeWithoutResult((action) → {
    roomMediaSessionDao.lockRoomId(upsertRMSRequest.getRoomId());

    int updated;
    updated = roomMediaSessionDao.updateByRoomSessionId(upsertRMSRequest);
    if (updated == 0) {
        updated = roomMediaSessionDao.updateCreatedRoomSession(upsertRMSRequest);
        if (updated == 0) {
            updated = roomMediaSessionDao.insertOrDoNothing(upsertRMSRequest);
        }
    }

    if (updated == 0) {
        throw new RuntimeException(String.format(
                "Failed to update info on room media sessions. 0 rows updated.
request: %s",
                upsertRMSRequest
        ));
    }
});
```

С локом в базе: реализация

```
transactionTemplate.executeWithoutResult((action) → {
    roomMediaSessionDao.lockRoomId(upsertRMSRequest.getRoomId());

    int updated;
    updated = roomMediaSessionDao.updateByRoomSessionId(upsertRMSRequest);
    if (updated == 0) {
        updated = roomMediaSessionDao.updateCreatedRoomSession(upsertRMSRequest);
        if (updated == 0) {
            updated = roomMediaSessionDao.insertOrDoNothing(upsertRMSRequest);
        }
    }

    if (updated == 0) {
        throw new RuntimeException(String.format(
                "Failed to update info on room media sessions. 0 rows updated.
request: %s",
                upsertRMSRequest
        ));
    }
});
```

С локом в базе: бенчмарки



Мы защитились!

Многопользовательские in-memory сессии больше не распадаются!

Что нам пригодилось:

- *Partial Index*
- *Разметка запросов для поиска в POWA*

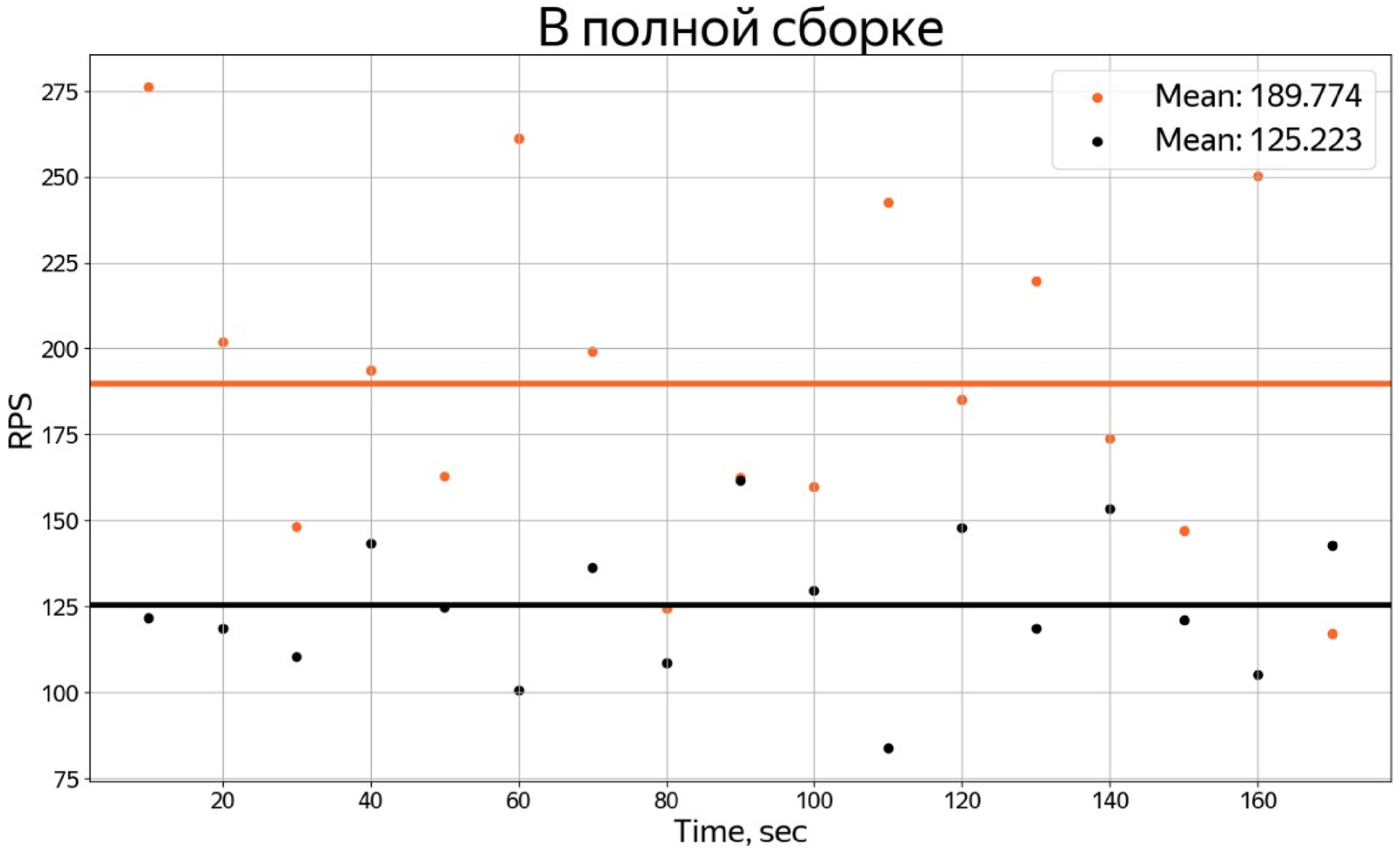
https://www.livemaster.ru/topic/888691-pritcha-grabli

Котаны занялись производитель-ностью

Работа без локов

```
int updated;
//point.1
updated = roomMediaSessionDao.updateByRoomSessionId(upsertRMSRequest);
if (updated == 0) {
    //point.2
    updated = roomMediaSessionDao.updateCreatedRoomSession(upsertRMSRequest);
    if (updated == 0) {
        //point.3
        updated = roomMediaSessionDao.insertOrDoNothing(upsertRMSRequest);
    }
}

if (updated == 0) {
    throw new RuntimeException(String.format(
            "Failed to update info on room media sessions. 0 rows updated. request: %s",
            upsertRMSRequest
    ));
}
```

Работа без локов: бенчмарки



Контрпример

```
int updated;
//point.1
updated = roomMediaSessionDao.updateByRoomSessionId(upsertRMSRequest);
if (updated == 0) {
    //point.2
    updated = roomMediaSessionDao.updateCreatedRoomSession(upsertRMSRequest);
    if (updated == 0) {
        //point.3 ← здесь параллельная сессия вставила другое состояние
        updated = roomMediaSessionDao.insertOrDoNothing(upsertRMSRequest);
    }
}

if (updated == 0) {
    throw new RuntimeException(String.format(
            "Failed to update info on room media sessions. 0 rows updated. request:
%s",
            upsertRMSRequest
    ));
}
```

По координатам (peer_id, rms_id)

Состояние	Присутствует запрошенный RMS ID	Присутствует created запись
BLANK	false	false
CREATED	false	true
RMSID	true	false
BOTH	true	true

По координатам: Blank

Состояние	Присутствует запрошенный RMS ID	Присутствует created запись
BLANK	false	false
CREATED	false	true
RMSID	true	false
BOTH	true	true

По координатам: Created

Состояние	Присутствует запрошенный RMS ID	Присутствует created запись
BLANK	false	false
CREATED	false	true
RMSID	true	false
BOTH	true	true

По координатам: RMSID

Состояние	Присутствует запрошенный RMS ID	Присутствует created запись
BLANK	false	false
CREATED	false	true
RMSID	true	false
BOTH	true	true

По координатам: BOTH

Состояние	Присутствует запрошенный RMS ID	Присутствует created запись
BLANK	false	false
CREATED	false	true
RMSID	true	false
BOTH	true	true

Постановка задачи

- Если до начала не было записи с rs_id, она появится

Состояние	Присутствует запрошенный RMS ID	Присутствует created запись
BLANK	false	false
CREATED	false	true
RMSID	true	false
BOTH	true	true

Постановка задачи

- *Если до начала не было записи с rs_id, она появится*
- *Если до начала была запись created, она станет записью с rs_id*

Состояние	Присутствует запрошенный RMS ID	Присутствует created запись
BLANK	false	false
CREATED	false	true
RMSID	true	false
BOTH	true	true

Постановка задачи

- Если до начала не было записи с rs_id, она появится
- Если до начала была запись created, она станет записью с rs_id
- Если до начала была запись с rs_id, в конце тоже будет запись с rs_id

Состояние	Присутствует запрошенный RMS ID	Присутствует created запись
BLANK	false	false
CREATED	false	true
RMSID	true	false
BOTH	true	true

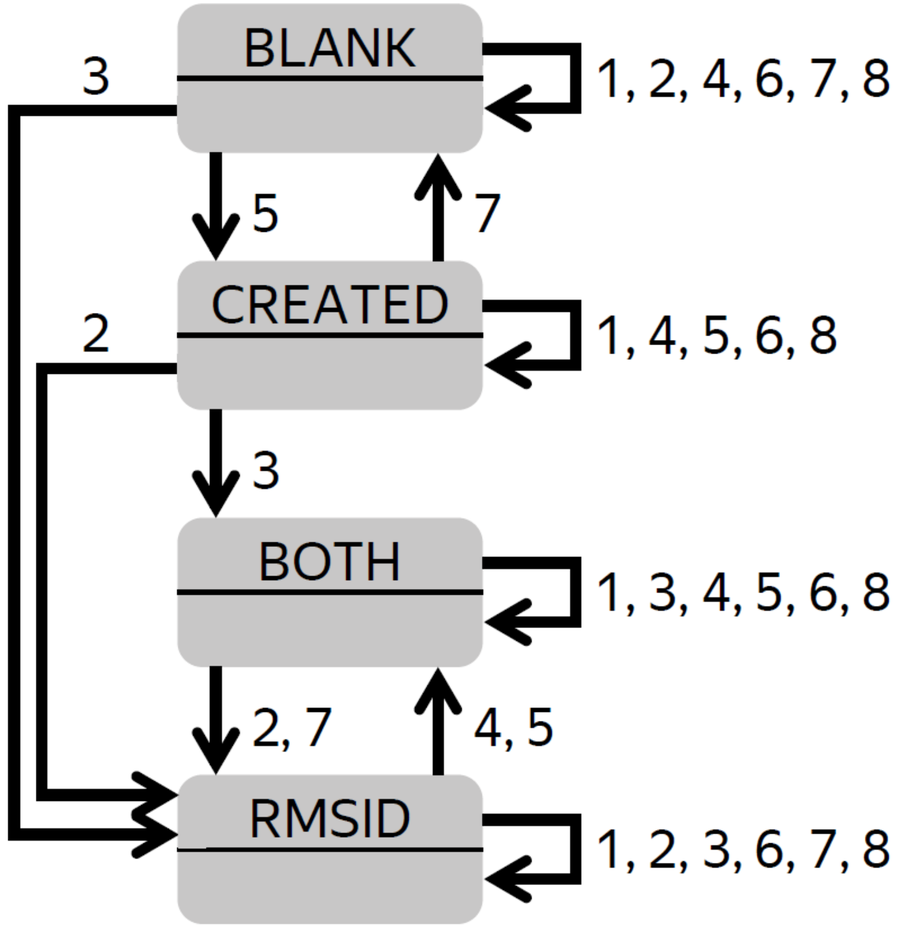
Постановка задачи

- Если до начала не было записи с rs_id, она появится
- Если до начала была запись created, она станет записью с rs_id
- Если до начала была запись с rs_id, в конце тоже будет запись с rs_id
- Новой created записи быть создано не должно

Состояние	Присутствует запрошенный RMS ID	Присутствует created запись
BLANK	false	false
CREATED	false	true
RMSID	true	false
BOTH	true	true

Работа без локов: конечный автомат

Диаграмма состояний



Update: match (peer_id, rs_id)

1 – с указанным RS
6 – с другим RS

Update записи (peer_id, rs_id)

Update: CREATED → Connected

1 – с указанным RS 6 – с другим RS	Update записи (peer_id, rs_id)
2 – с указанным RS 7 – с другим RS	Update записи (peer_id, RMS_UNKNOWN) → (peer_id, rs_id)

Update: insert новой записи

1 – с указанным RS 6 – с другим RS	Update записи (peer_id, rs_id)
2 – с указанным RS 7 – с другим RS	Update записи (peer_id, RMS_UNKNOWN) → (peer_id, rs_id)
3 – с указанным RS 8 – с другим RS	Insert записи (peer_id, rs_id)

Update: disconnect

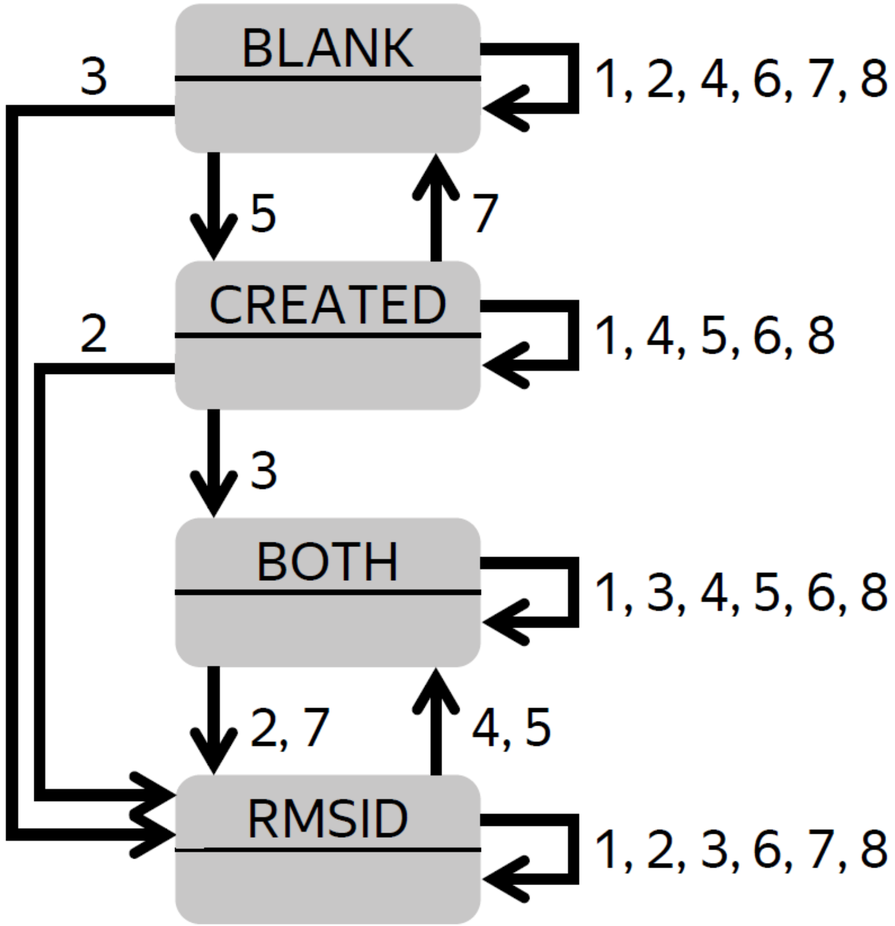
1 – с указанным RS 6 – с другим RS	Update записи (peer_id, rs_id)
2 – с указанным RS 7 – с другим RS	Update записи (peer_id, RMS_UNKNOWN) → (peer_id, rs_id)
3 – с указанным RS 8 – с другим RS	Insert записи (peer_id, rs_id)
4	Disconnect: Update записи (peer_id, rs_id) + insert (peer_id, created)

Update: insert CREATED записи

1 – с указанным RS 6 – с другим RS	Update записи (peer_id, rs_id)
2 – с указанным RS 7 – с другим RS	Update записи (peer_id, RMS_UNKNOWN) → (peer_id, rs_id)
3 – с указанным RS 8 – с другим RS	Insert записи (peer_id, rs_id)
4	Disconnect: Update записи (peer_id, rs_id) + insert (peer_id, created)
5	GetConnection: Insert (peer_id, RMS_UNKNOWN)

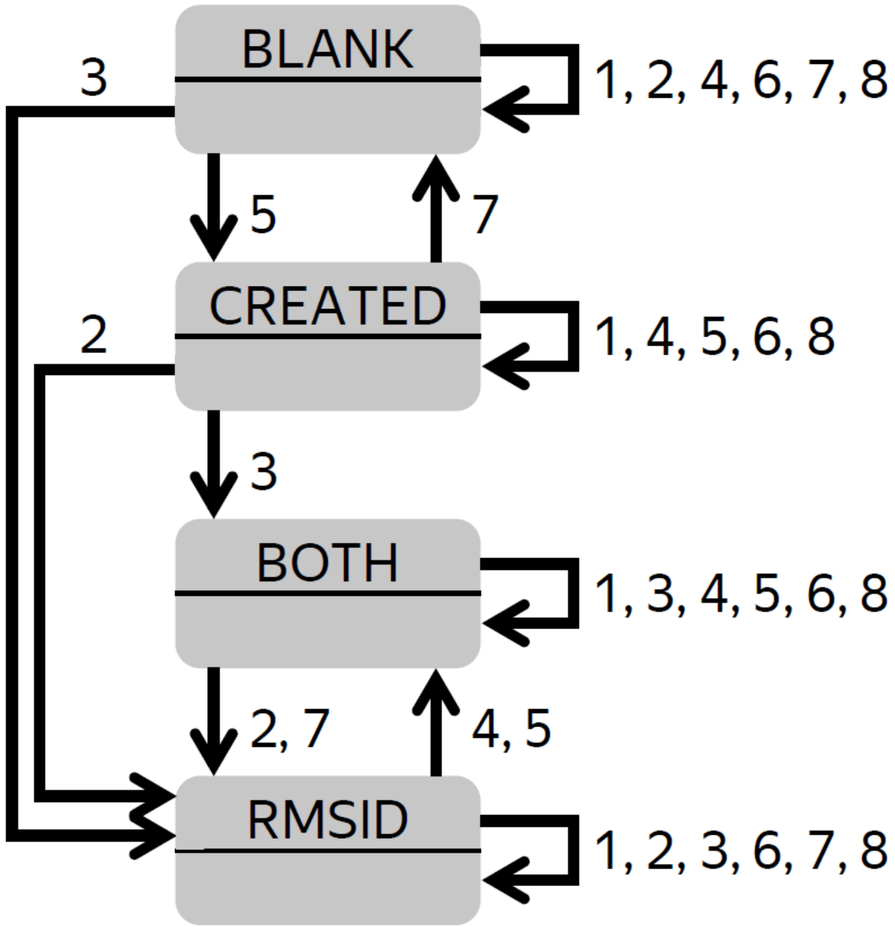
Работа без локов: анализ

Диаграмма состояний



Работа без локов: анализ

Диаграмма состояний



Последовательность команд, участвующих в гонках

Upsert своего RMSID	*1, 2, 3, 1*
Upsert чужого RMSID	*6, 7, 8, 6*
disco	*4*
getConnection	*5*

Идея доказательства

- *В команде от медиа сервера всегда есть Room Session ID*
- *За 1 шаг всегда можно досоздать запись или перевести ее из CREATED*
- *В разных положениях шаг разный*
- *Эволюция положения в обратном порядке невозможна*

Unit Tests: 100%

Комбинаторный взрыв

Наличие / отсутствие Created сессии со своим room session id

Состояние Room Media Session другого пира с другим room session id

1. *Archived*
2. *Created*
3. *First Offer*
4. *Connected*
5. *Disconnected*

Состояние Room Media Session в базе

1. *Archived*
2. *First Offer*
3. *Connected*
4. *Отсутствует*
5. *Disconnected*

Виды команд:

1. *FIRST_OFFER*
2. *CONNECTED*
3. *DISCONNECTED*

Итого:
*5*5*3*2 = **150***

Unit Tests: 100%

Мы посчитали достаточным

Проверить влияние на других пиров только для FIRST_OFFER

Проверить Created → FIRST_OFFER только для команды FIRST_OFFER

Состояние Room Media Session в базе

1. *Archived*
2. *CREATED*
3. *First Offer*
4. *Connected*
5. *Отсутствует*
6. *Disconnected*

Виды команд:

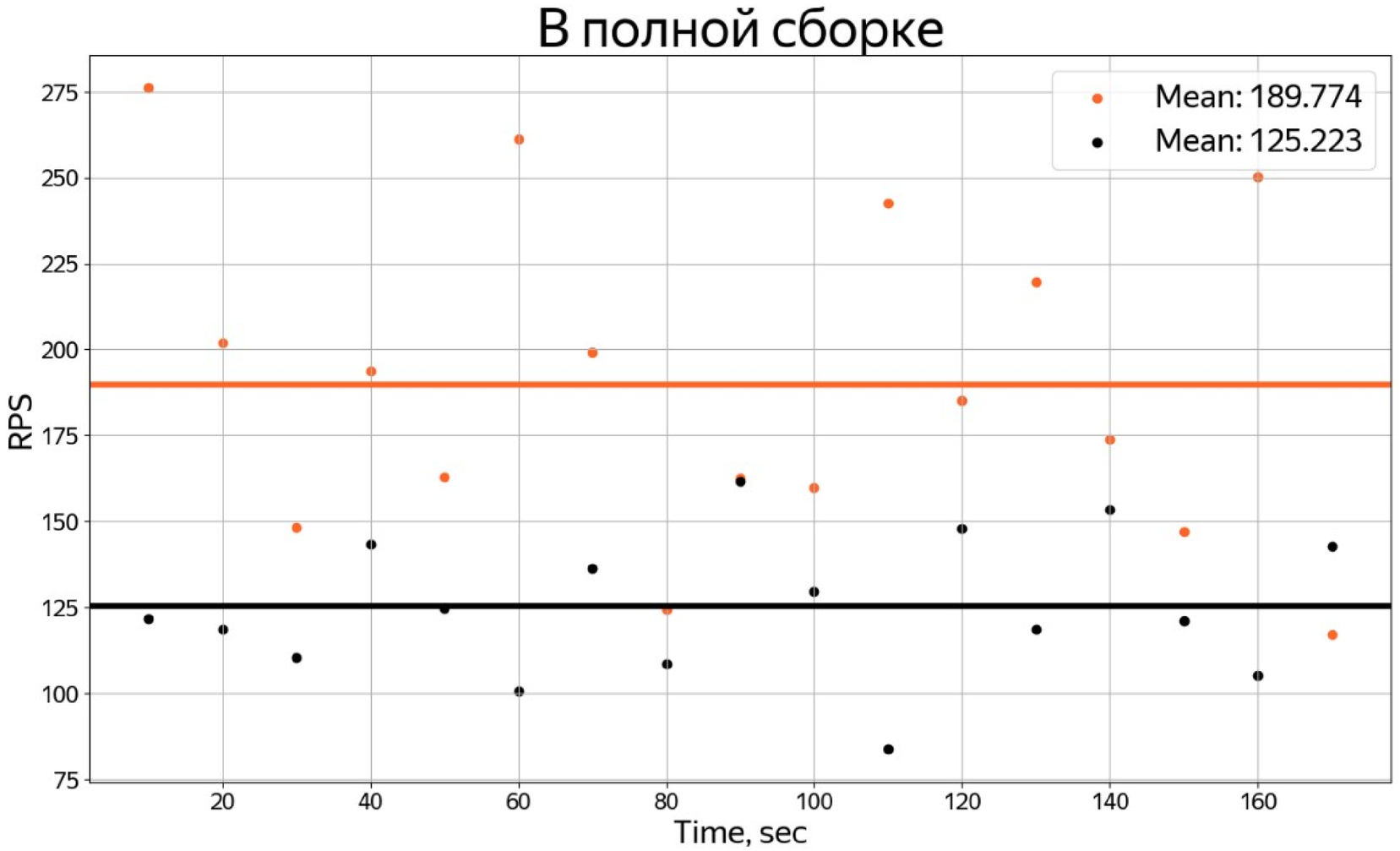
1. *FIRST_OFFER*
2. *CONNECTED*
3. *DISCONNECTED*

Итого:
*5 + 1 + 6*3 = 24*

Работа без локов: правильное решение

```
int updated;
//point.1
updated = roomMediaSessionDao.updateByRoomSessionId(upsertRMSRequest);
if (updated == 0) {
    //point.2
    updated = roomMediaSessionDao.updateCreatedRoomSession(upsertRMSRequest);
    if (updated == 0) {
        //point.3
        updated = roomMediaSessionDao.insertOrDoNothing(upsertRMSRequest);
        //point.4
        if (updated == 0) {
            updated = roomMediaSessionDao.updateByRoomSessionId(upsertRMSRequest);
        }
    }
}
```

Работа без локов: правильное решение



Сравним на стенде

01

Docker-Compose

02

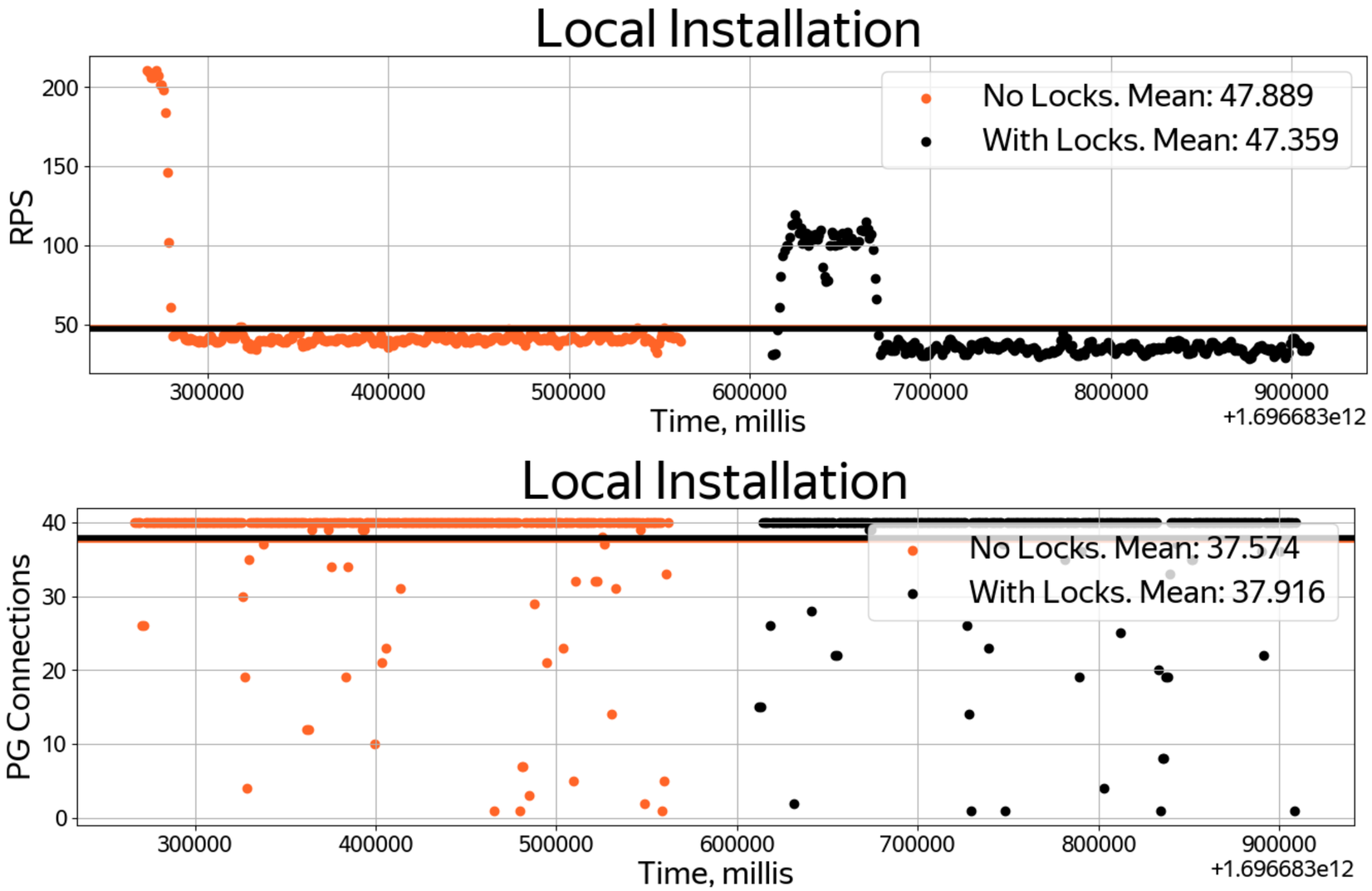
Postgers:
4 CPU cores

03

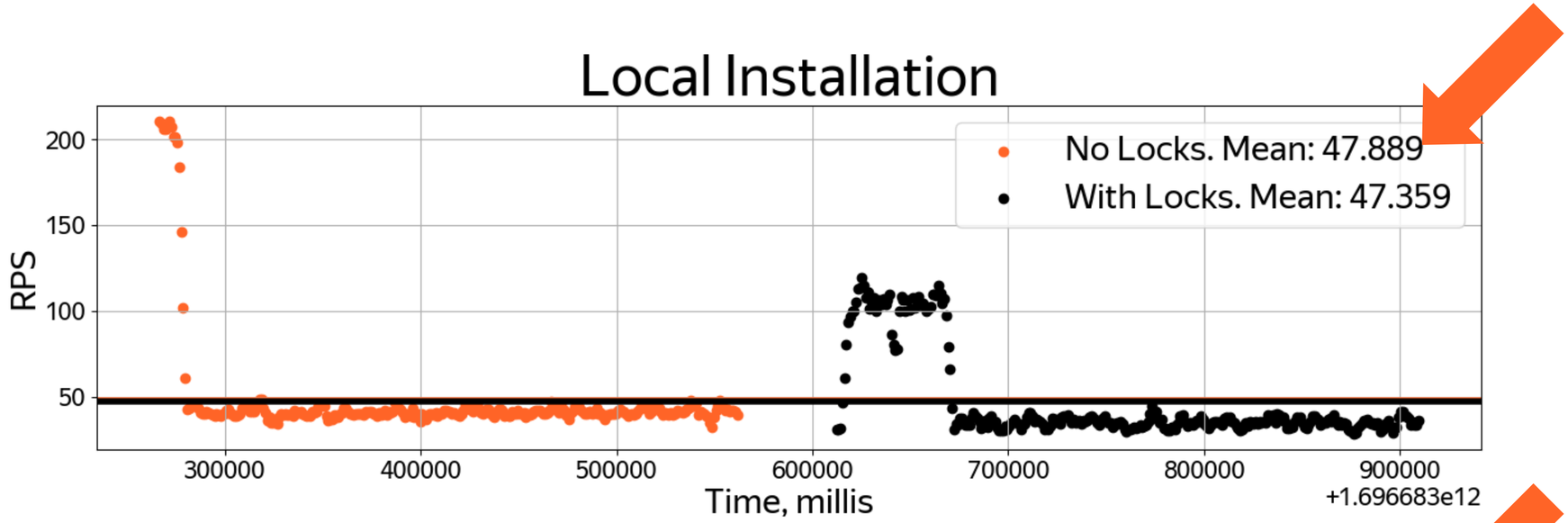
Full RPS /
Throttled RPS

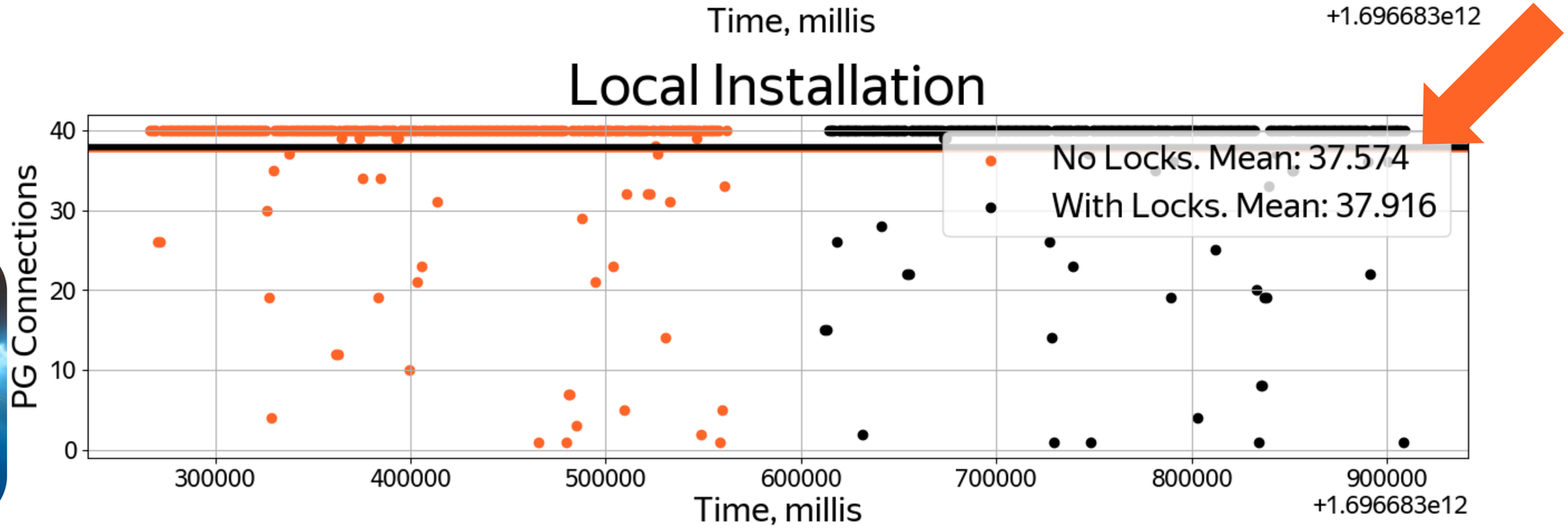
Котаны собрали стенд

С локами и без локов на стенде



С локами и без локов на стенде





Эмулируем задержки сети

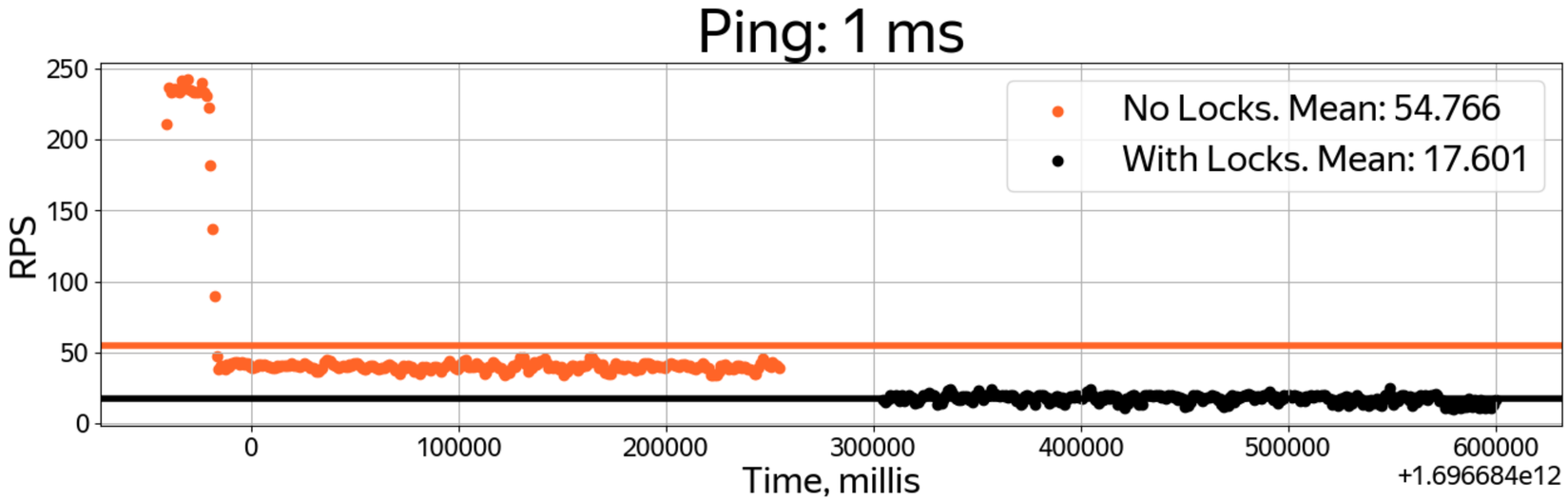
```
#Docker-compose.yml
services:
  postgres:
    container_name: postgres
    cap_add:
      - NET_ADMIN
```

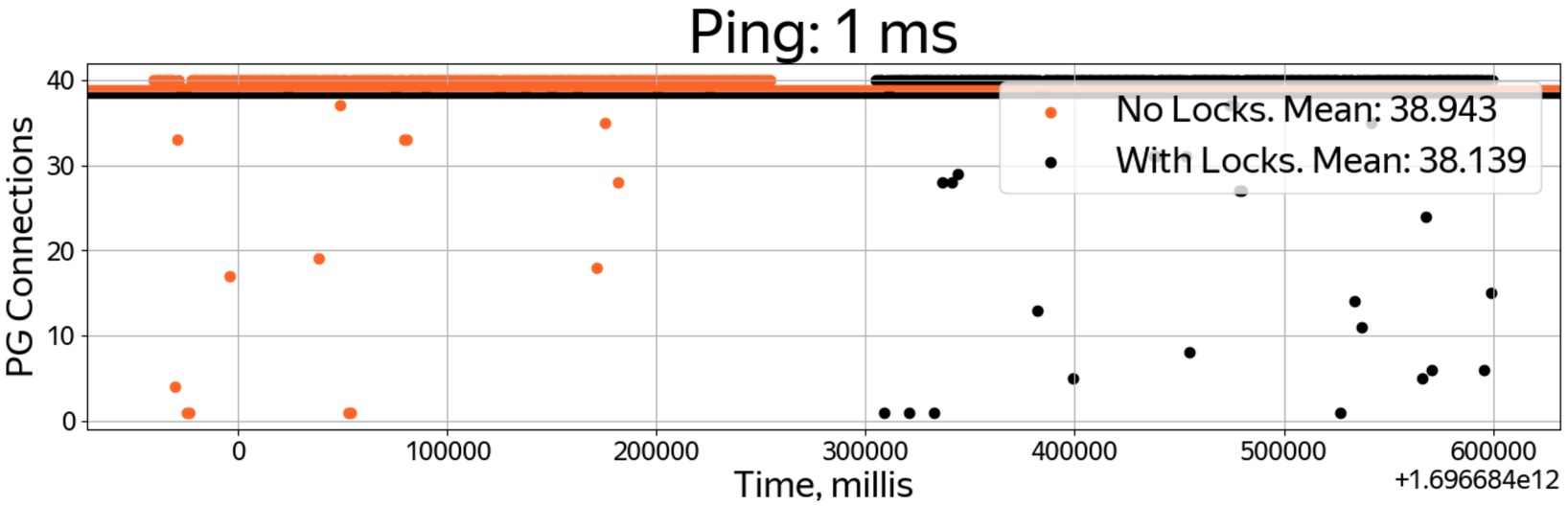
```
#!/bin/bash

echo "setting PG delay to $1"

docker exec postgres tc qdisc del dev eth0 root netem delay 1ms

docker exec postgres tc qdisc add dev eth0 root netem delay "$1"
```

Ознакомиться со стендом

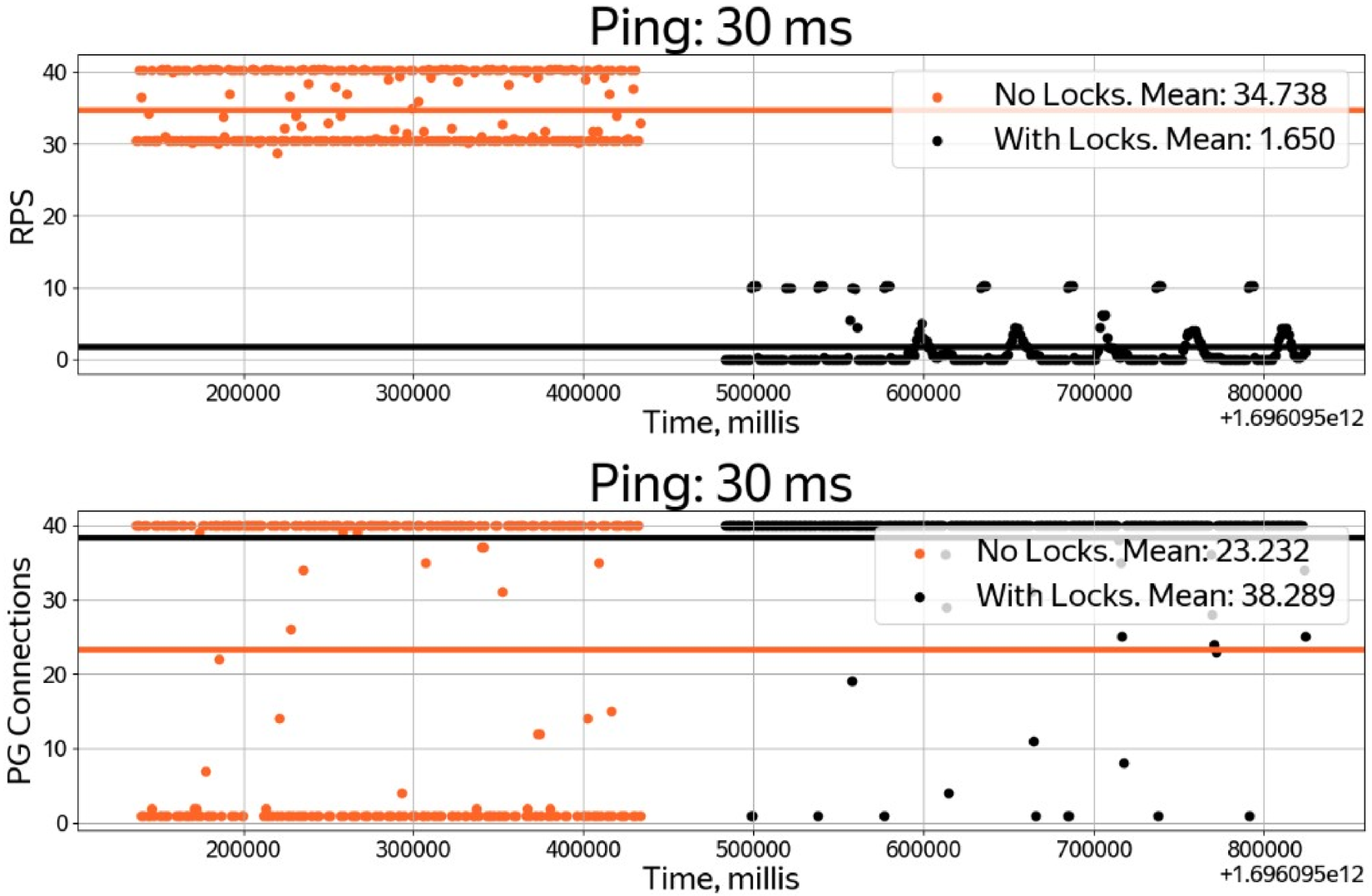
https://github.com/topright007/tmost_sm_bench

На стенде с задержкой

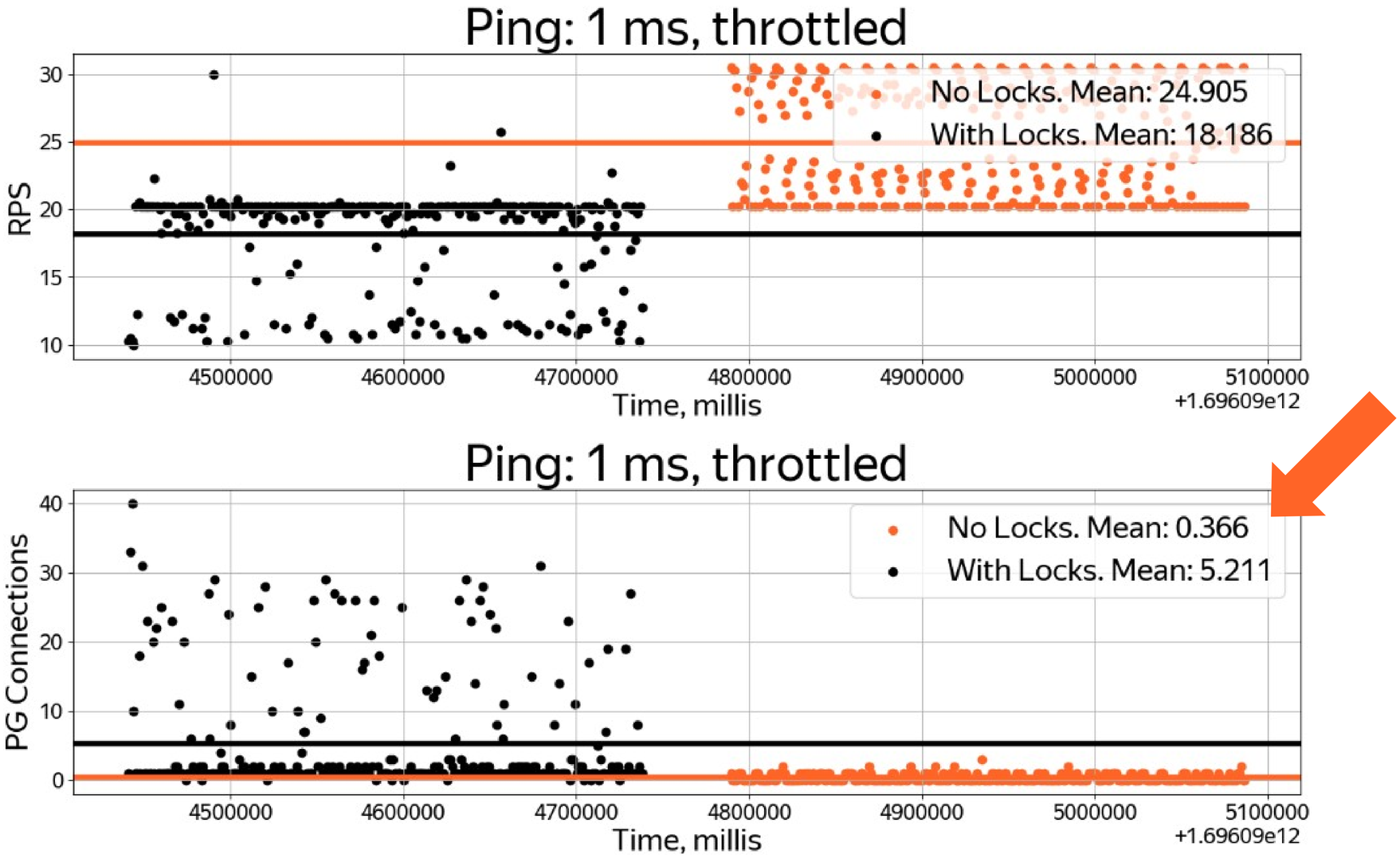




На стенде с задержкой 30 ms



Когда RPS не в полку



Мы ускорились!

Что нам пригодилось:

- *Конечный Автомат*
- *Экономия PG Connections*

Что нам помешало:

- *Смена PK*
  (peer_id, UNKNOWN_RMS_ID)
  → (peer_id, rs_id)

Limor Zellermayer, unsplash.com

Слайды

https://disk.yandex.ru/d/Vuaiz3N-5nF6GA

Яндекс 360 на Joker

Итоги & Q&A

- *Stateful In-memory компоненты без репликации at scale — очень хочется. Можно!*
- *Менять Primary Key в многопоточном окружении — больно. Но можно*
- *Транзакции — дорого.*
  И не всегда нужно

Дмитрий Некрылов topright@yandex-team.ru *TG: @topright007*

 https://kartinkof.club/jumor/prikolnie/4699-prikolnye-kartinki-pro-poklon-44-foto.html

Яндекс 360

Спасибо!

Дмитрий Некрылов topright@yandex-team.ru *TG: @topright007*