




ОНЛАЙН-ОБРАЗОВАНИЕ

Онлайн-образование



Меня хорошо видно && слышно?

Ставьте  , если все хорошо
Напишите в чат, если есть проблемы
заодно проверяем, включена ли запись занятия

Включил Юджин запись ли ты





Журналы



ЗОЛОТОВ АНТОН

telegram @AVZolotov

Правила вебинара



Активно участвуем



Задаем вопрос в чат



Вопросы вижу в чате, могу ответить не сразу

Маршрут вебинара

Буферный кэш



Журнал предзаписи



Контрольная точка



Настройки журнала

Цели вебинара | После занятия вы сможете

1 Настраивать журналирование

2 Корректно настроить схему контрольных точек

Смысл | Зачем вам это уметь

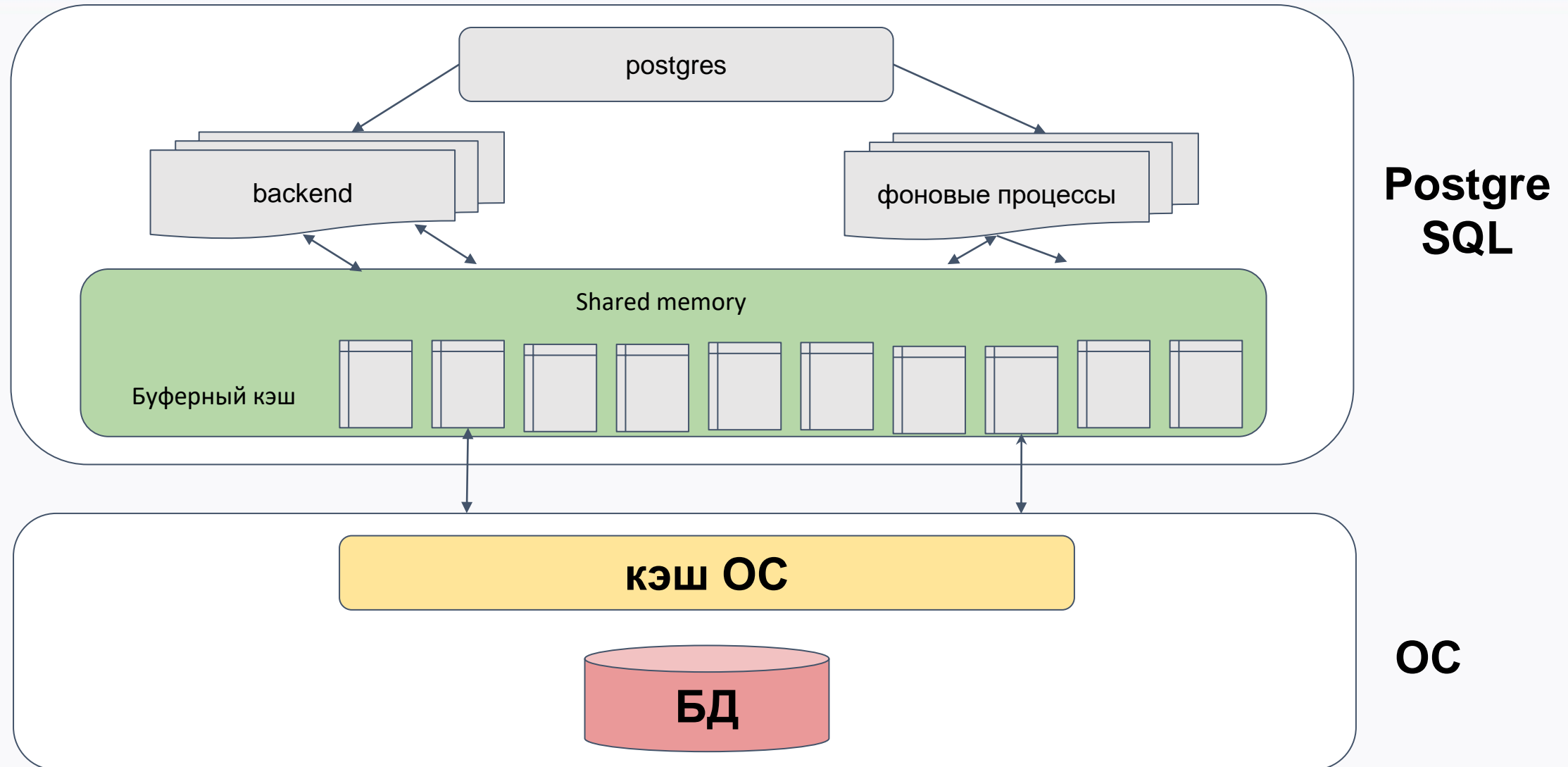
1 Обеспечить высокую надежность

2 Обеспечить оптимальную
производительность

The background of the slide is an aerial photograph of a dense city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue layer. A network of thin, light blue lines connects various points across the blue area, creating a web-like pattern. The text "Буферный кэш" is centered in white, bold, sans-serif font.

Буферный кэш

Буферный кэш



Буферный кэш. Зачем?

Ускоряем работу всей системы.

- Оперативная память очень быстра, но ее мало.
- ЖД огромный, но медленный.

Буферный кэш. Состав

Каждый буфер состоит из одной страницы данных и заголовка. Размер по умолчанию 8 кб. Заголовок содержит:

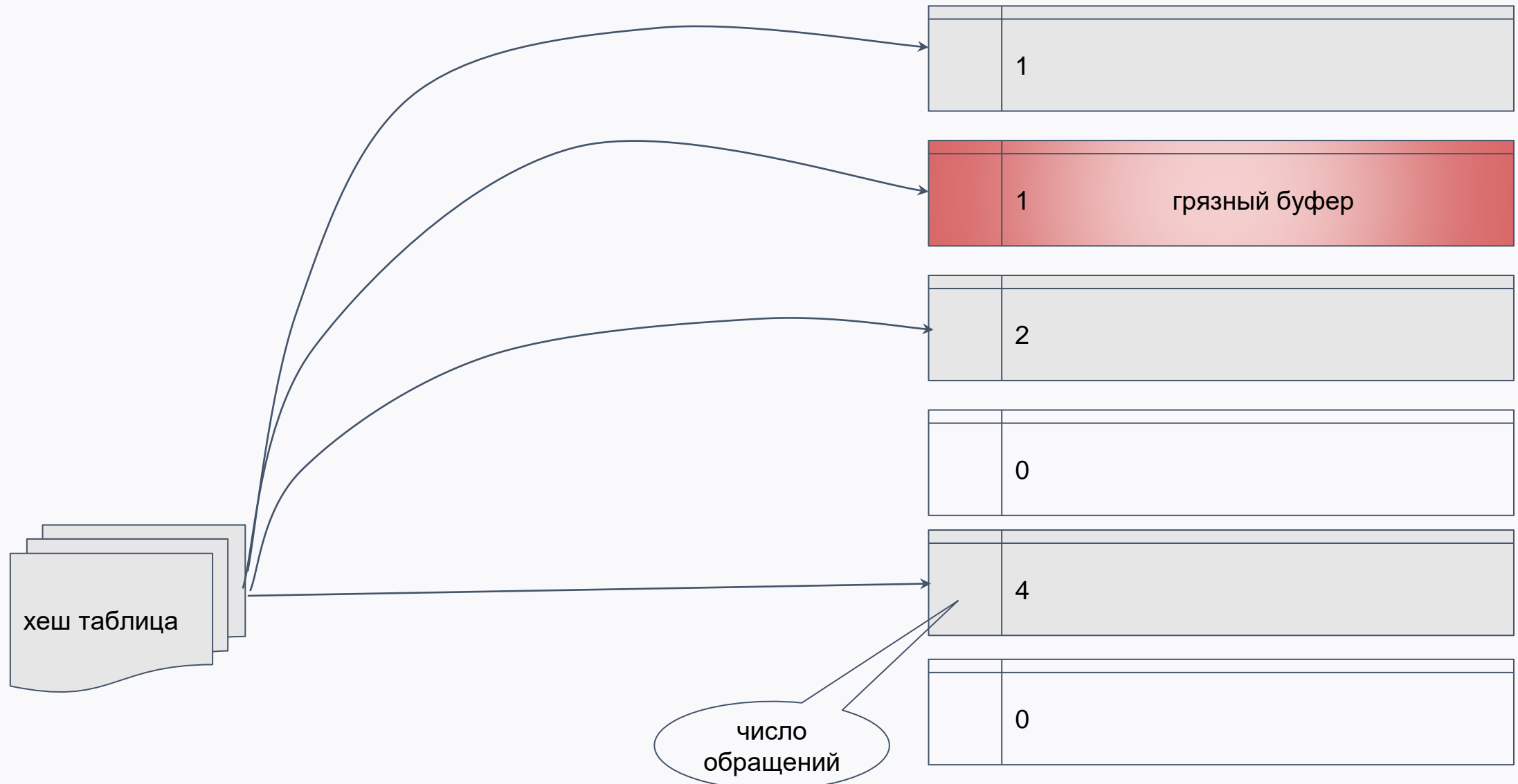
- расположение страницы на диске (файл и номер страницы в нем),
- число обращений к буферу (счетчик увеличивается каждый раз, когда процесс читает или изменяет буфер, максимально значение 5),
- признак того, что данные на странице изменились и рано или поздно должны быть записаны на диск (грязный буфер).

Изначально кэш содержит:

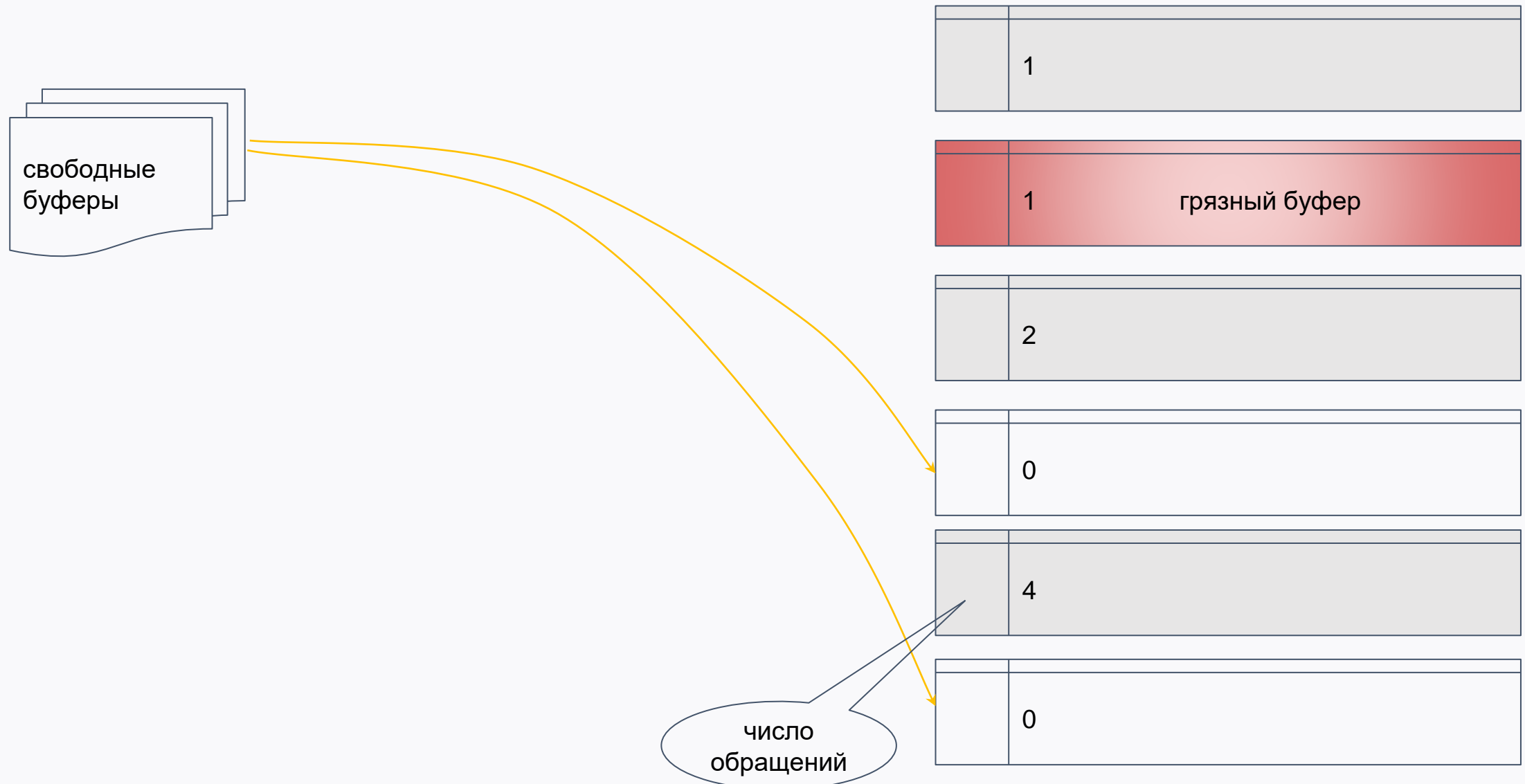
- пустые буферы, и все они связаны в список свободных буферов,
- указатель на «следующую жертву» при вытеснении старых буферов,
- также используется хеш-таблица, чтобы быстро находить нужную страницу в кэше.

Размер буферного кэша задается параметром **shared_buffers**. Его изменение **требует перезапуска** сервера.

Буферный кэш. Механизм работы

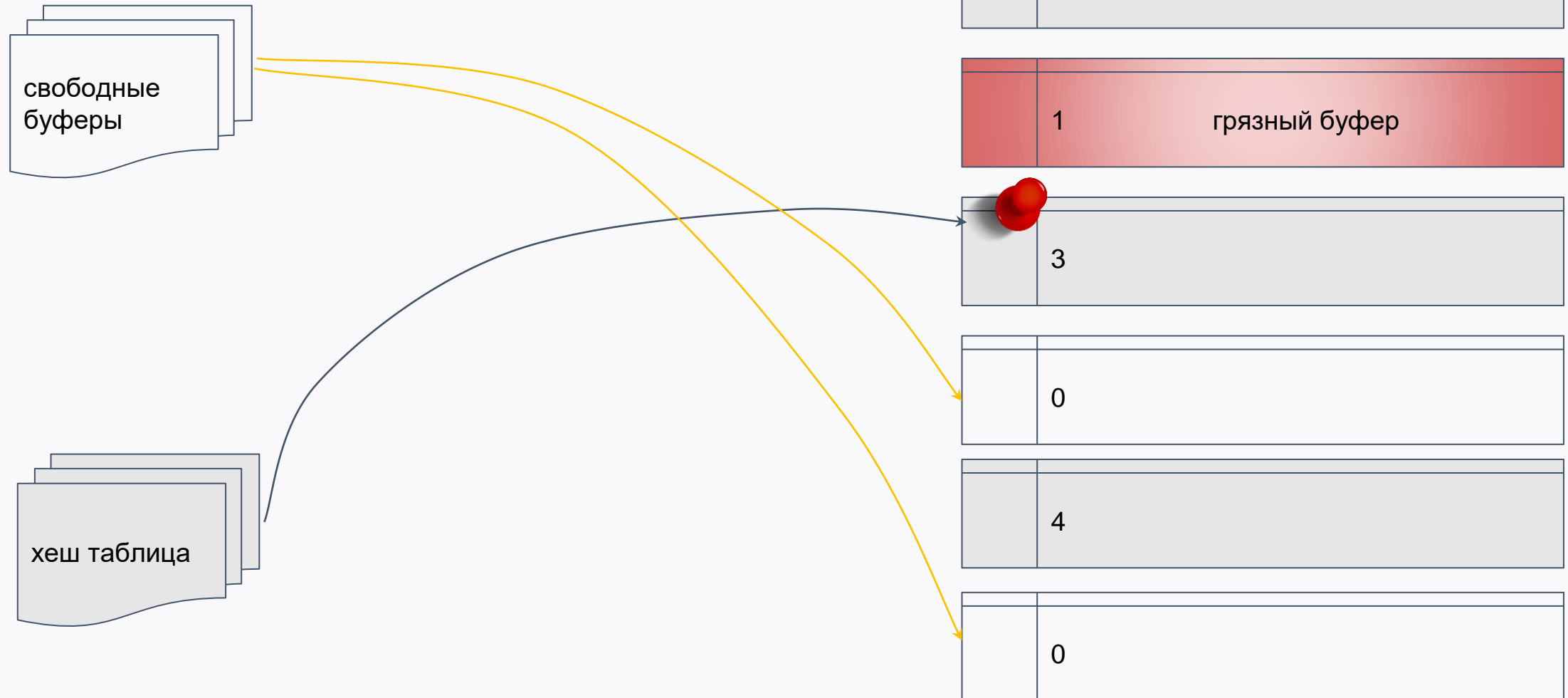


Буферный кэш. Механизм работы



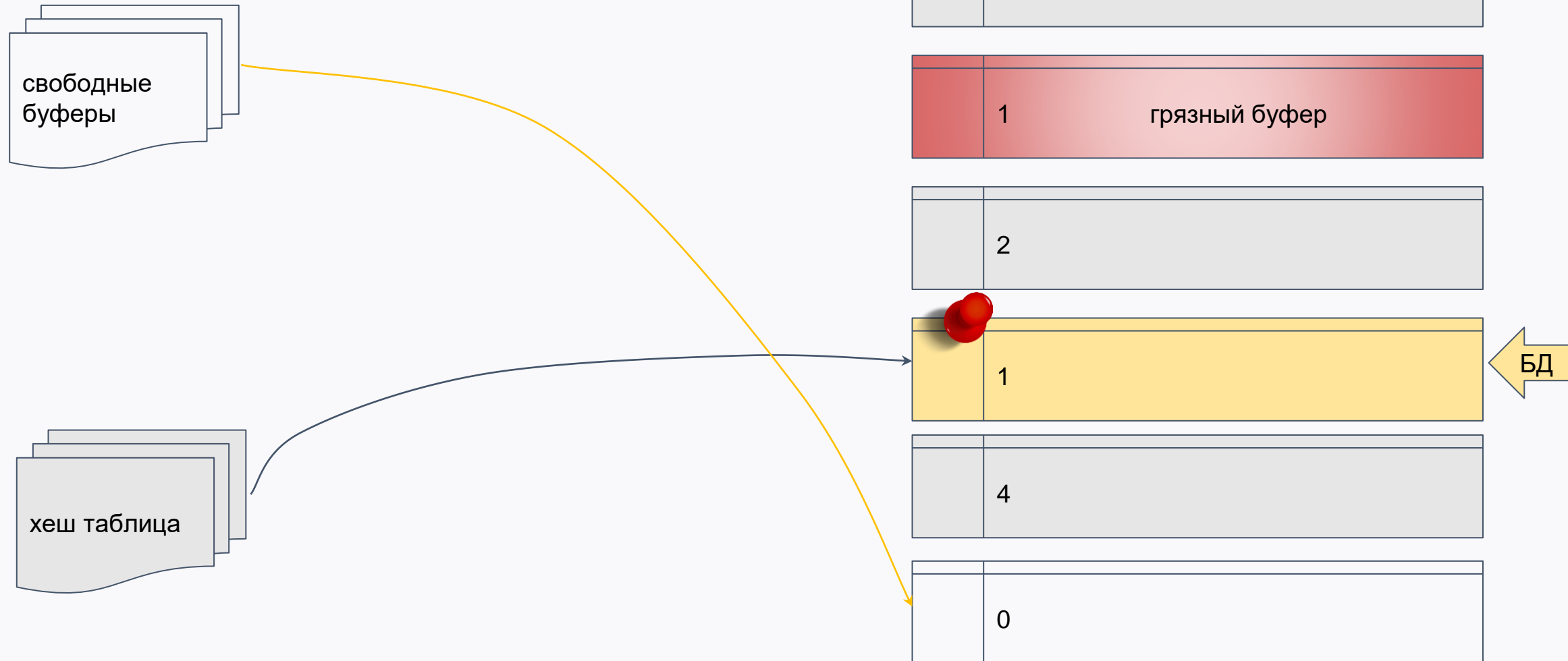
Буферный кэш. Механизм работы

сначала ищем в буферном кэше по хешу. Если нужная страница найдена в кэше, процесс должен «закрепить» буфер (увеличить счетчик pin count) и увеличить число обращений(счетчик usage count)



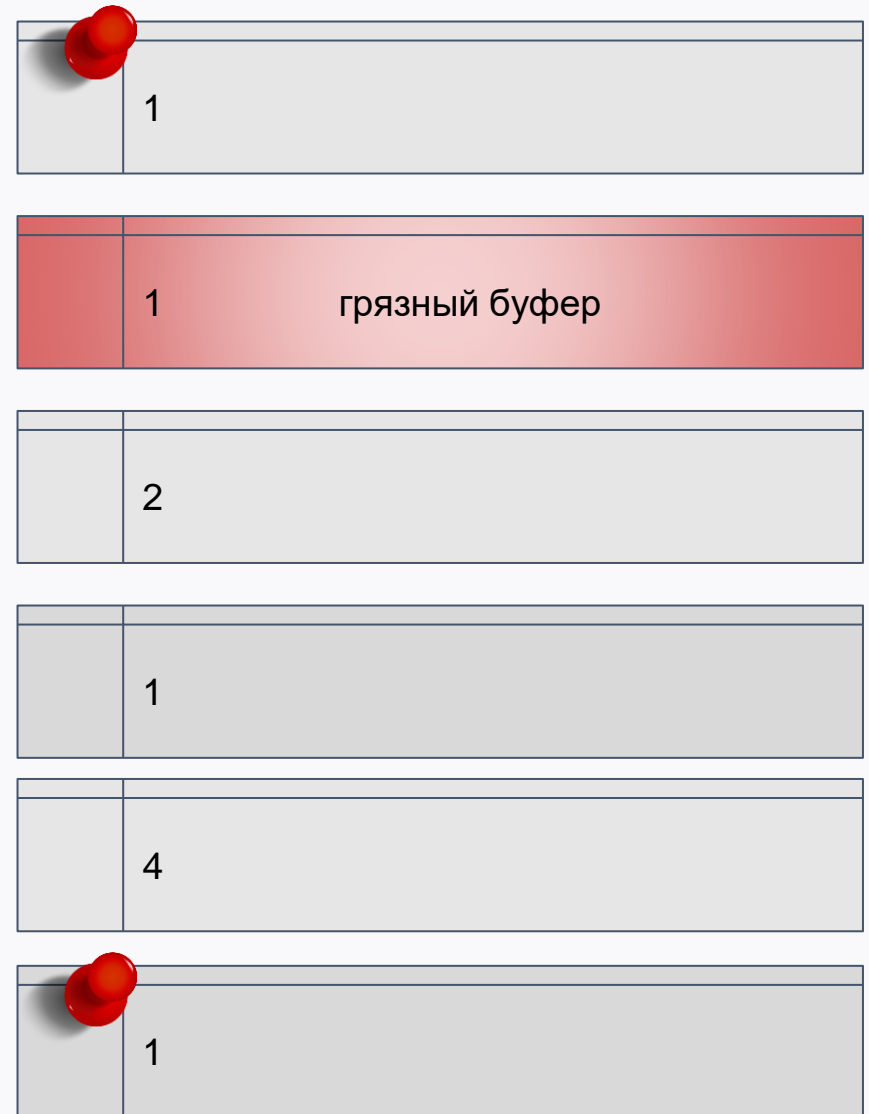
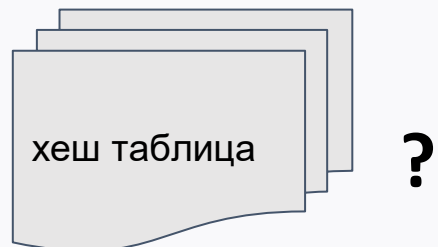
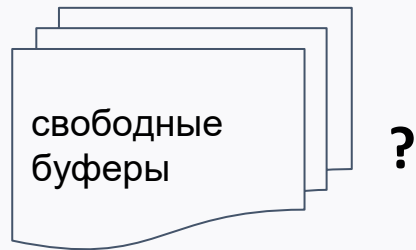
Буферный кэш. Механизм работы

если нет страницы в кэше, то грузим с диска в первый пустой блок и добавляем строку в хеш таблицу



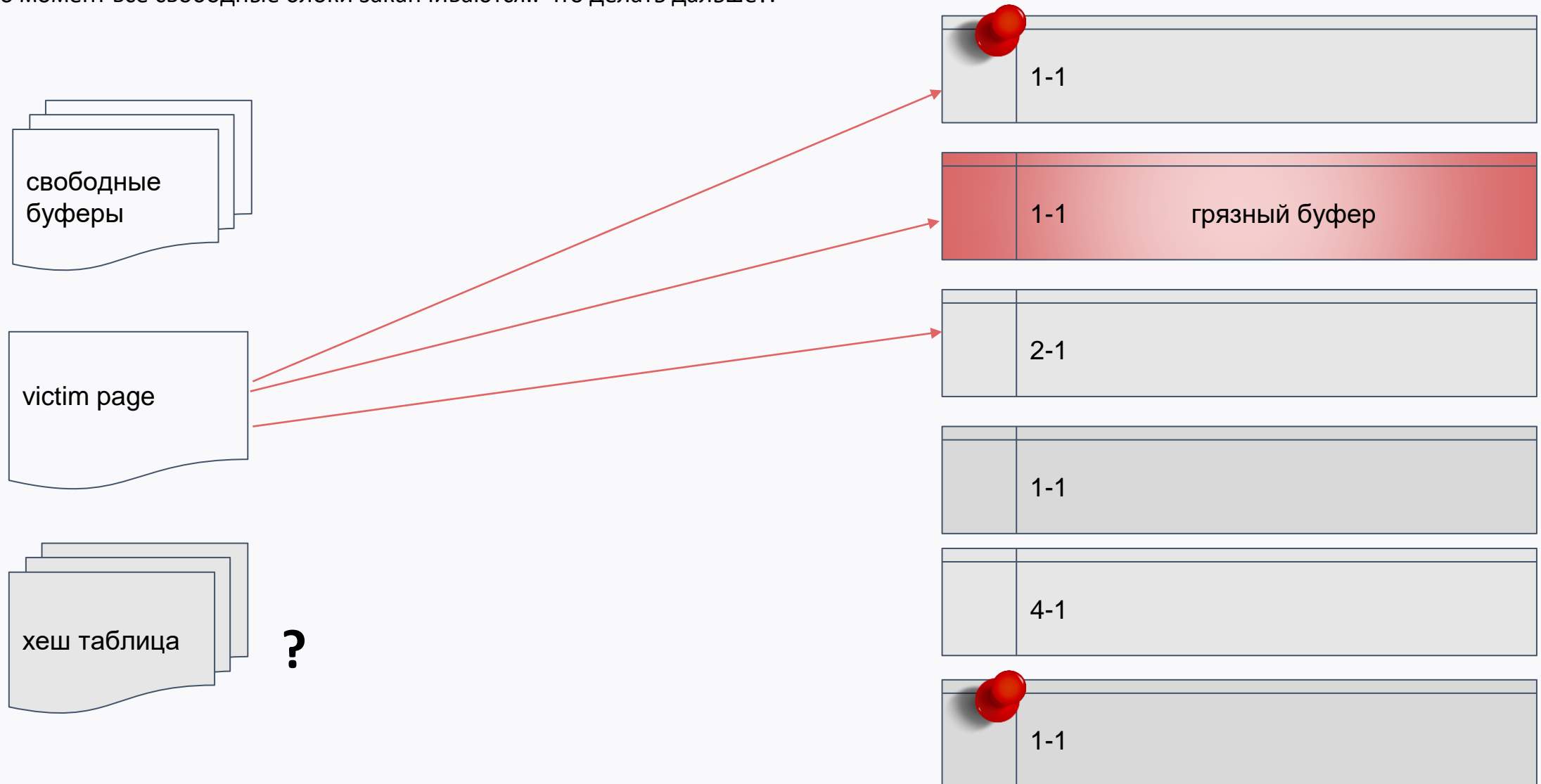
Буферный кэш. Чтение с вытеснением

в какой-то момент все свободные блоки заканчиваются.. Что делать дальше?



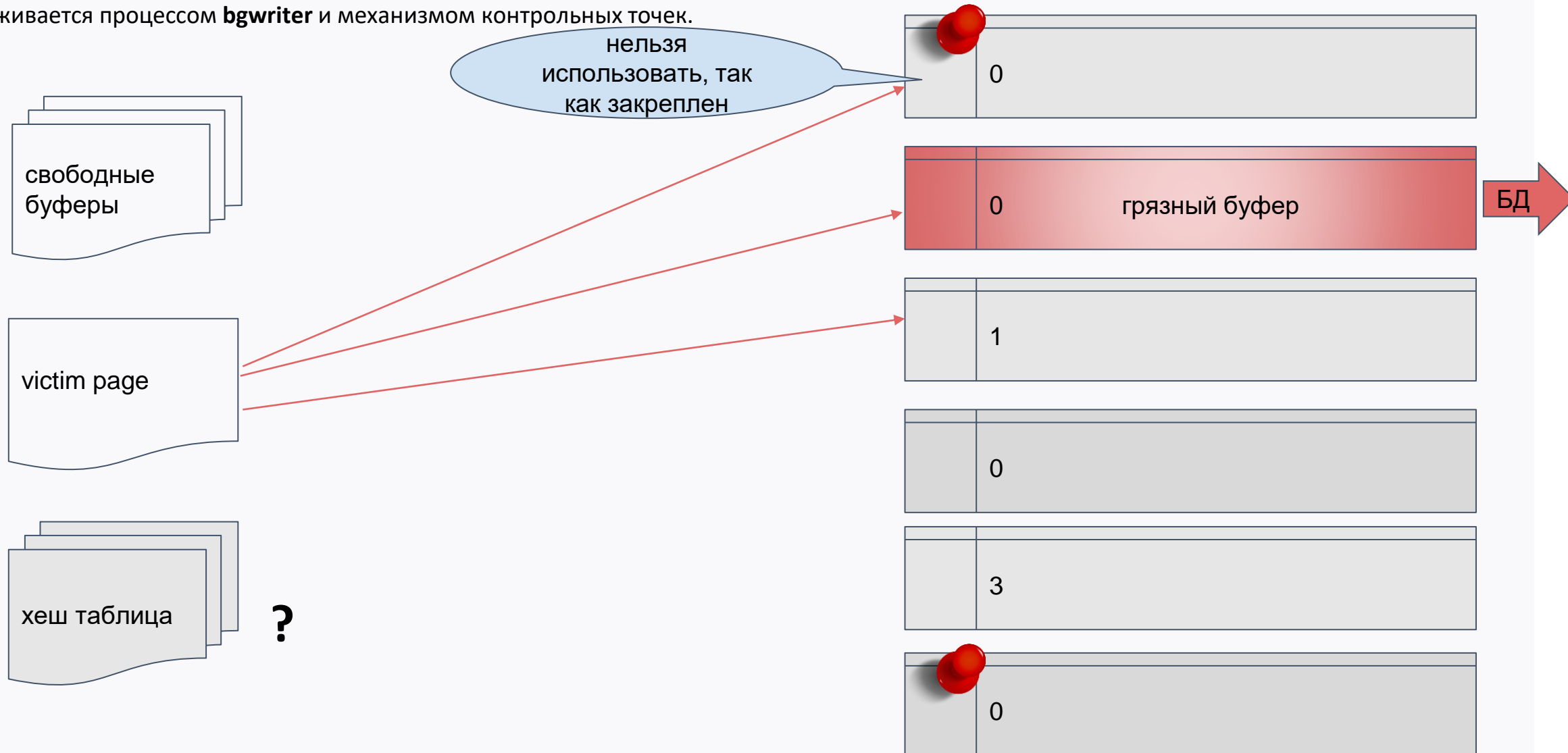
Буферный кэш. Чтение с вытеснением

в какой-то момент все свободные блоки заканчиваются.. Что делать дальше?.

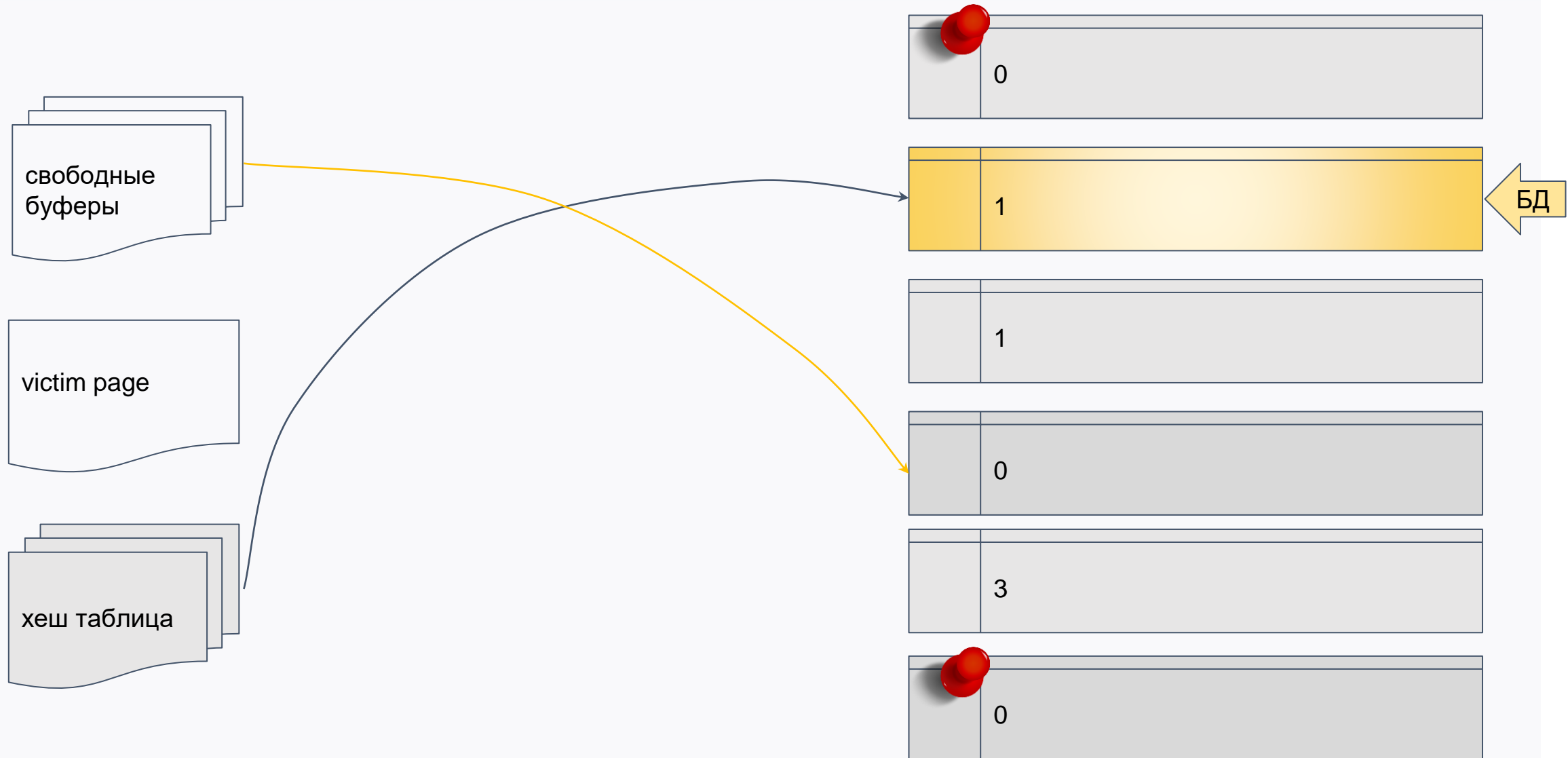


Буферный кэш. Чтение с вытеснением

Разную страницу нужно **заменить**, но для этого ее нужно **сбросить на диск**.
частично сглаживается процессом **bgwriter** и механизмом контрольных точек.



Буферный кэш. Чтение с вытеснением



Буферный кэш. Массовое вытеснение

Буферное кольцо

часть буферного кэша, выделенная для одной операции предотвращает вытеснение кэша «одноразовыми» данными

операция	кол-во страниц	грязные буферы
последовательное чтение (несколько операций одновременно)	32	исключаются из кольца
очистка (VACUUM)	32	вытесняются на диск
массовая запись (COPY, CTAS)	≤ 2048	вытесняются на диск

Буферный кэш. Настройка

По умолчанию **shared_buffers = 128MB**

Буферный кэш должен содержать «активные» данные:

- при меньшем размере постоянно вытесняются полезные страницы
- при большем размере бессмысленно растут накладные расходы

начальная рекомендация — **25%** ОЗУ

Нужно учитывать двойное кэширование - если страницы нет в кэше СУБД, она может оказаться в кэше ОС, но алгоритм вытеснения ОС не учитывает специфики базы данных.

Буферный кэш. Временные таблицы

Данные временных таблиц

- видны только одному сеансу — нет смысла использовать общий кэш
- существуют в пределах сеанса — не жалко потерять при сбое

Используется локальный буферный кэш

- не требуются блокировки
- память выделяется по необходимости в пределах `temp_buffers`
- обычный алгоритм вытеснения

Буферный кэш. Разогрев кэша

- **pg_prewarm**
- используется после рестарта кластера
- заполняет кэш указанными таблицами

Буферный кэш. Практика

практика

[11: 65.4. Карта видимости : Компания Postgres Professional](#)

[Postgres Pro Standard : Документация: 12: 65.3. Карта свободного пространства](#)

[WAL в PostgreSQL: 1. Буферный кэш / Блог компании Postgres Professional / Хабр](#)

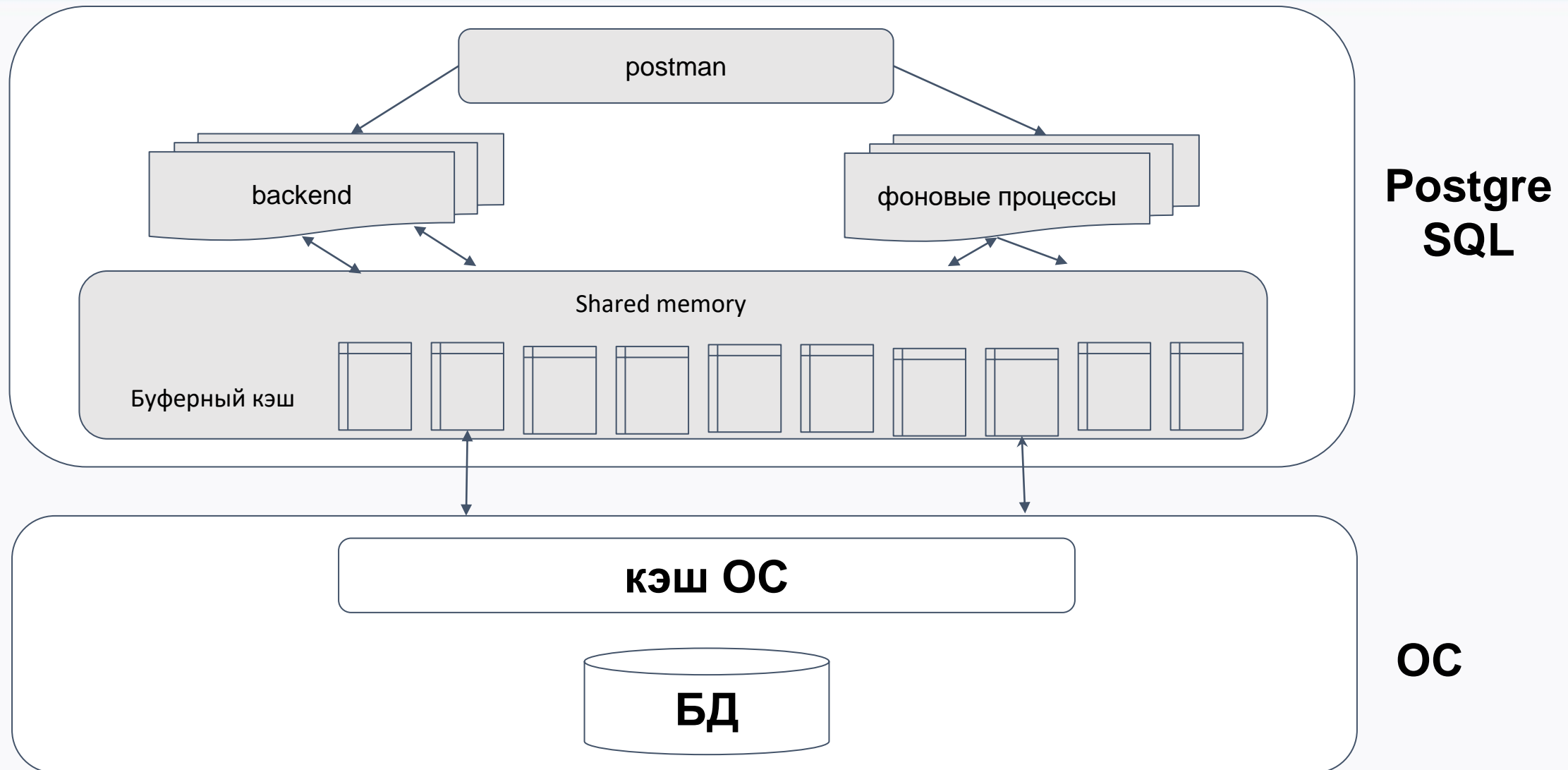
The image features a high-angle, aerial view of a dense urban skyline, likely New York City, with numerous skyscrapers and buildings. The entire image is overlaid with a semi-transparent blue and green gradient. A network of thin, light blue lines connects various points across the image, creating a digital or technological feel. The text "Вопросы?" is centered in the middle of the image in a large, white, sans-serif font.

Вопросы?

The background of the image is an aerial photograph of a dense urban skyline, likely New York City, with numerous skyscrapers. The entire image is overlaid with a semi-transparent blue layer. A network of thin, light blue lines connects various points across the blue area, creating a digital or data network aesthetic. The text "Журнал предзаписи" is centered in the middle of the image in a bold, white, sans-serif font.

Журнал предзаписи

Буферный кэш & WAL



Write ahead log - WAL

Основная задача

- ВОЗМОЖНОСТЬ ВОССТАНОВЛЕНИЯ СОГЛАСОВАННОСТИ ДАННЫХ ПОСЛЕ СБОЯ

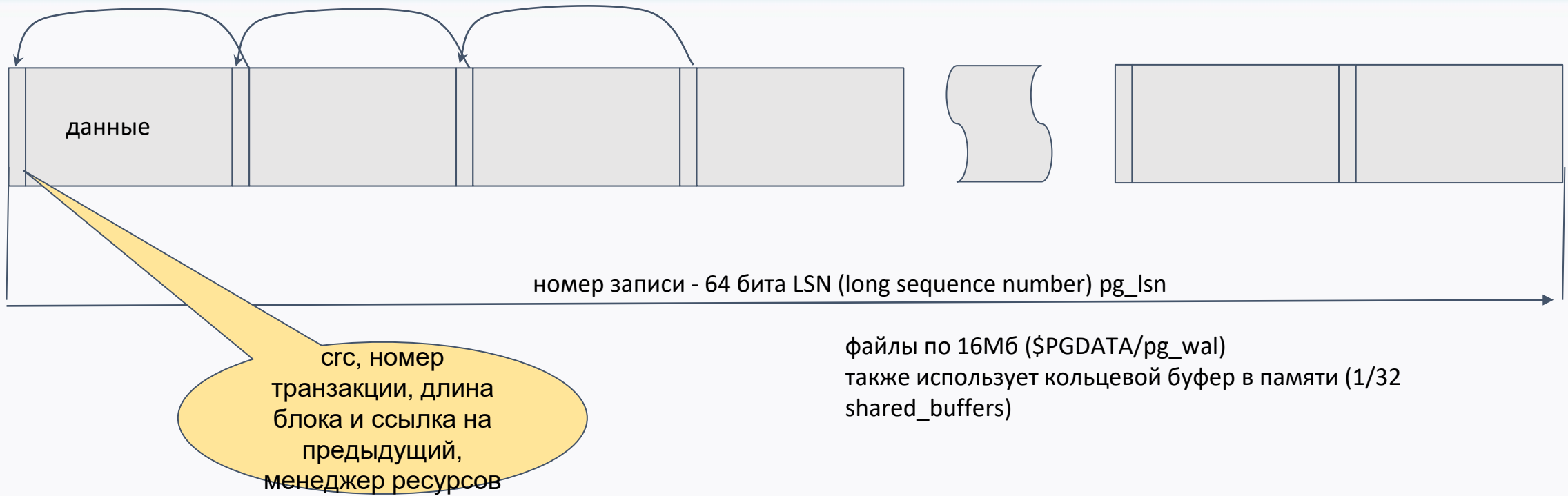
Механизм

- при изменении данных действие также записывается в журнал
журнальная запись попадает на диск раньше измененных данных
- восстановление после сбоя — повторное выполнение потерянных операций с помощью журнальных записей

Что туда попадает

- изменение любых страниц в буферном кэше
- фиксация и отмена транзакций - буферы ХАСТ
- НЕ ПОПАДАЮТ - временные и нежурналируемые таблицы

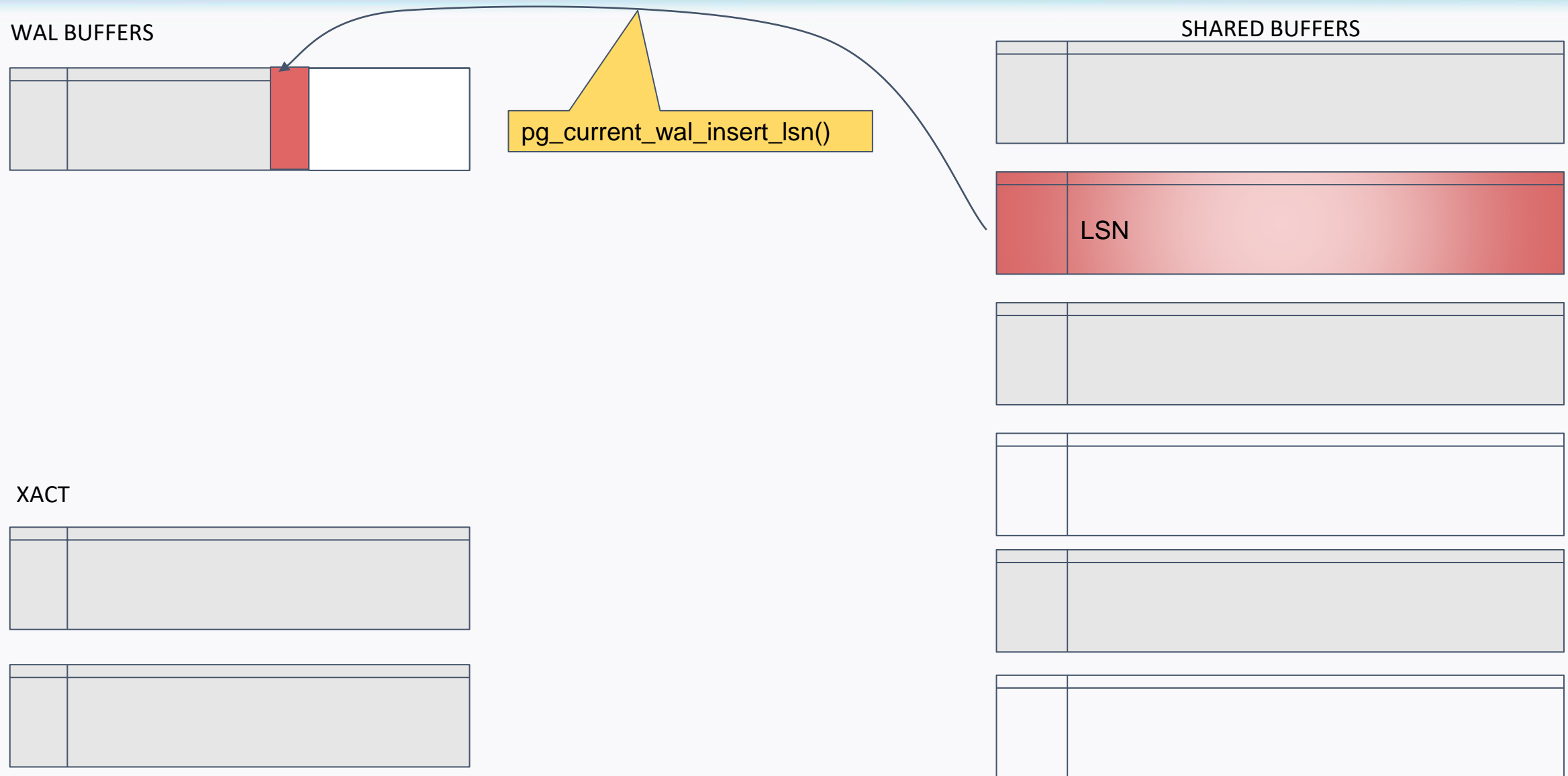
WAL. Устройство



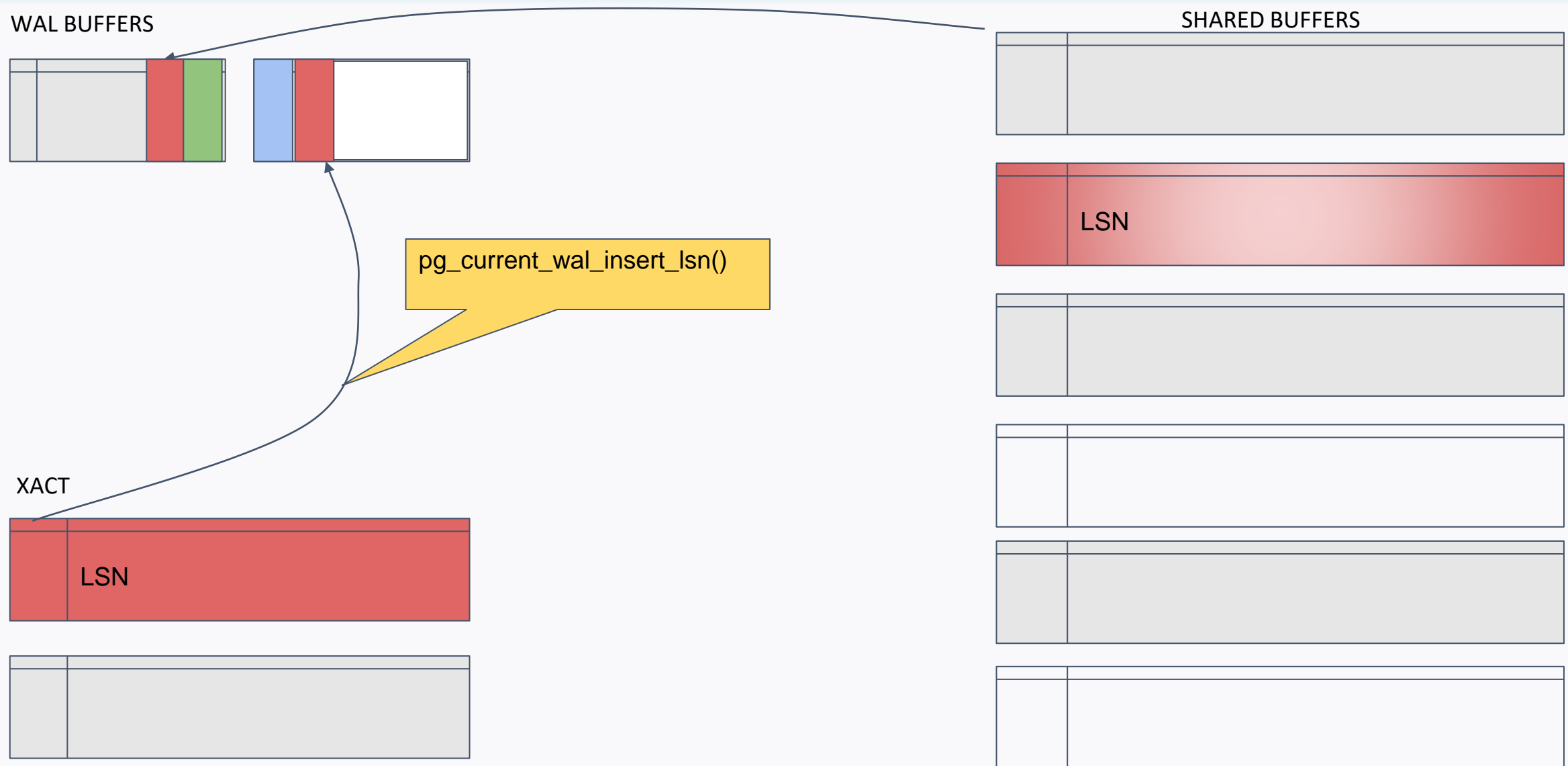
файлы по 16Мб (\$PGDATA/pg_wal)
также использует кольцевой буфер в памяти (1/32
shared_buffers)

`$/usr/lib/postgresql/13/bin/pg_waldump -r list` - список менеджеров

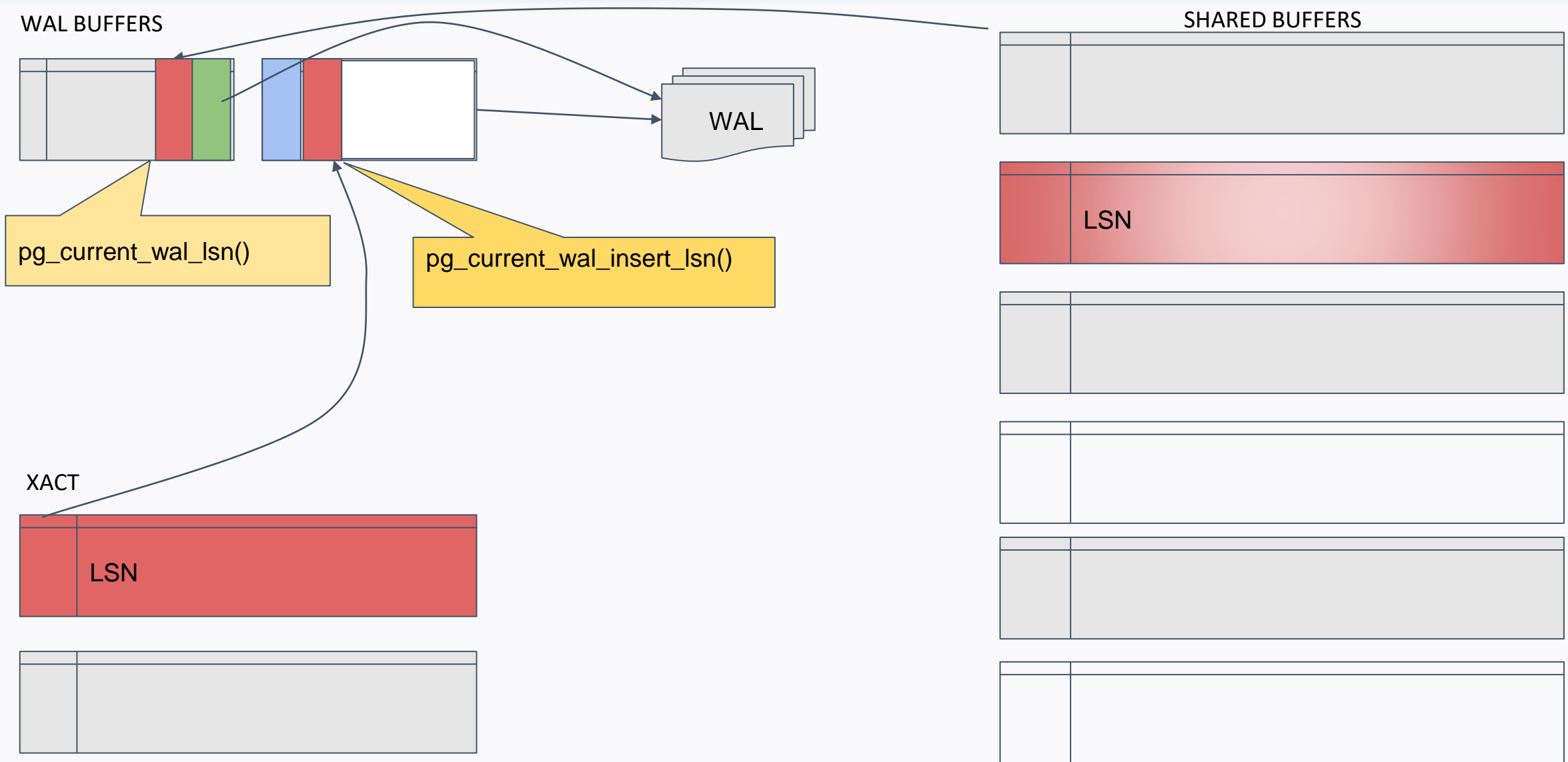
WAL. Механизм упреждающей записи



WAL. Механизм упреждающей записи

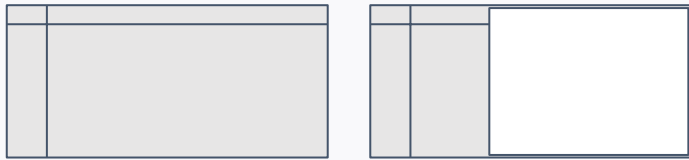


WAL. Механизм упреждающей записи

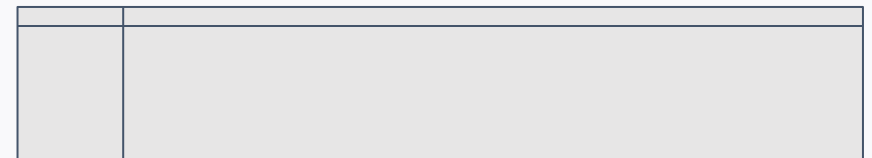


WAL. Механизм упреждающей записи

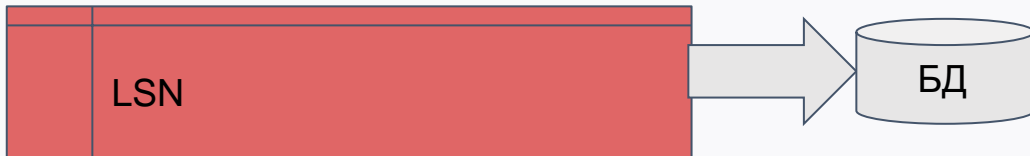
WAL BUFFERS



SHARED BUFFERS



ХАСТ



WAL. Восстановление

при старте сервера после сбоя

(состояние кластера в pg_control отличается от «shut down»):

1. для каждой журнальной записи:

- определить страницу, к которой относится эта запись
- применить запись, если ее LSN больше, чем LSN страницы

2. перезаписать нежурналируемые таблицы init-файлами

The image features a background of a dense city skyline, likely New York City, viewed from an elevated perspective. The entire image is overlaid with a semi-transparent blue and green gradient. A network of thin, light blue lines connects various points across the image, creating a digital or technological feel. The text "Вопросы?" is centered in the middle of the image in a large, white, sans-serif font.

Вопросы?

The image features a high-angle, aerial view of a dense urban skyline, likely New York City, with numerous skyscrapers and buildings. The entire image is overlaid with a semi-transparent blue layer. A network of thin, light blue lines connects various points across the blue area, creating a digital or data network aesthetic. The text 'Контрольная точка' is centered in the middle of the image in a bold, white, sans-serif font.

Контрольная точка

Контрольная точка

Зачем она нужна?

можно же с самого начала накатить все wal?

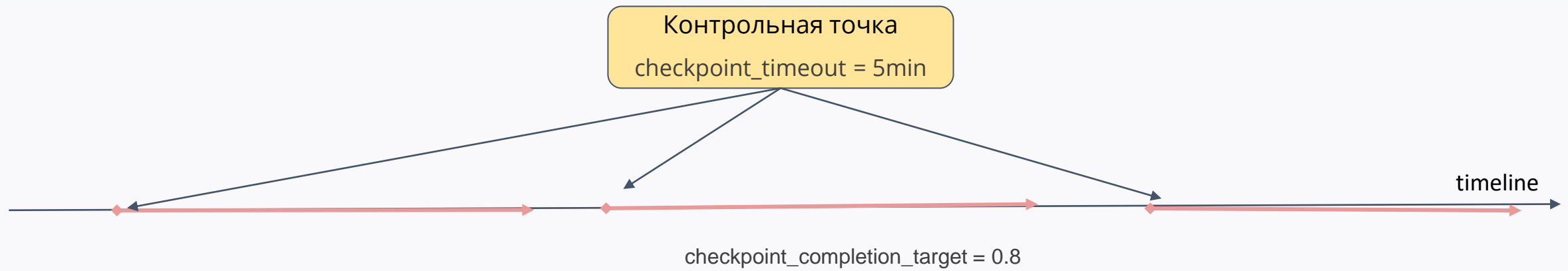
Контрольная точка

Зачем она нужна?

можно же с самого начала накатить все wal

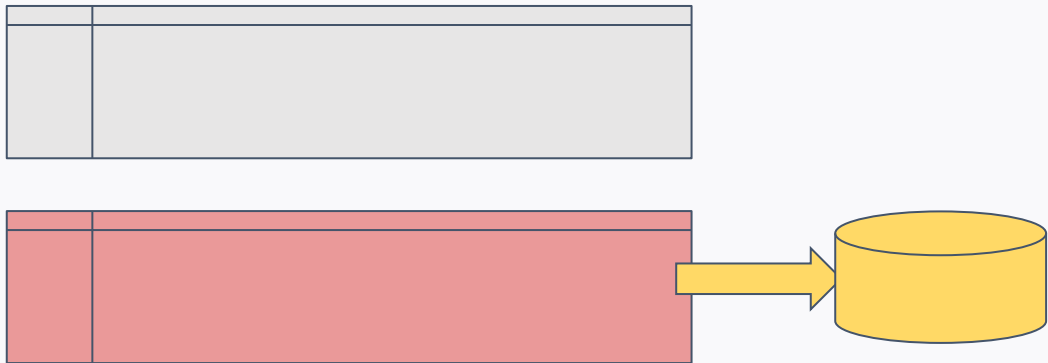
- очень большой объем информации хранить
- большое время восстановления
- сколько может страница измененная лежать в буферном кэше?

Контрольная точка



Контрольная точка

ХАСТ



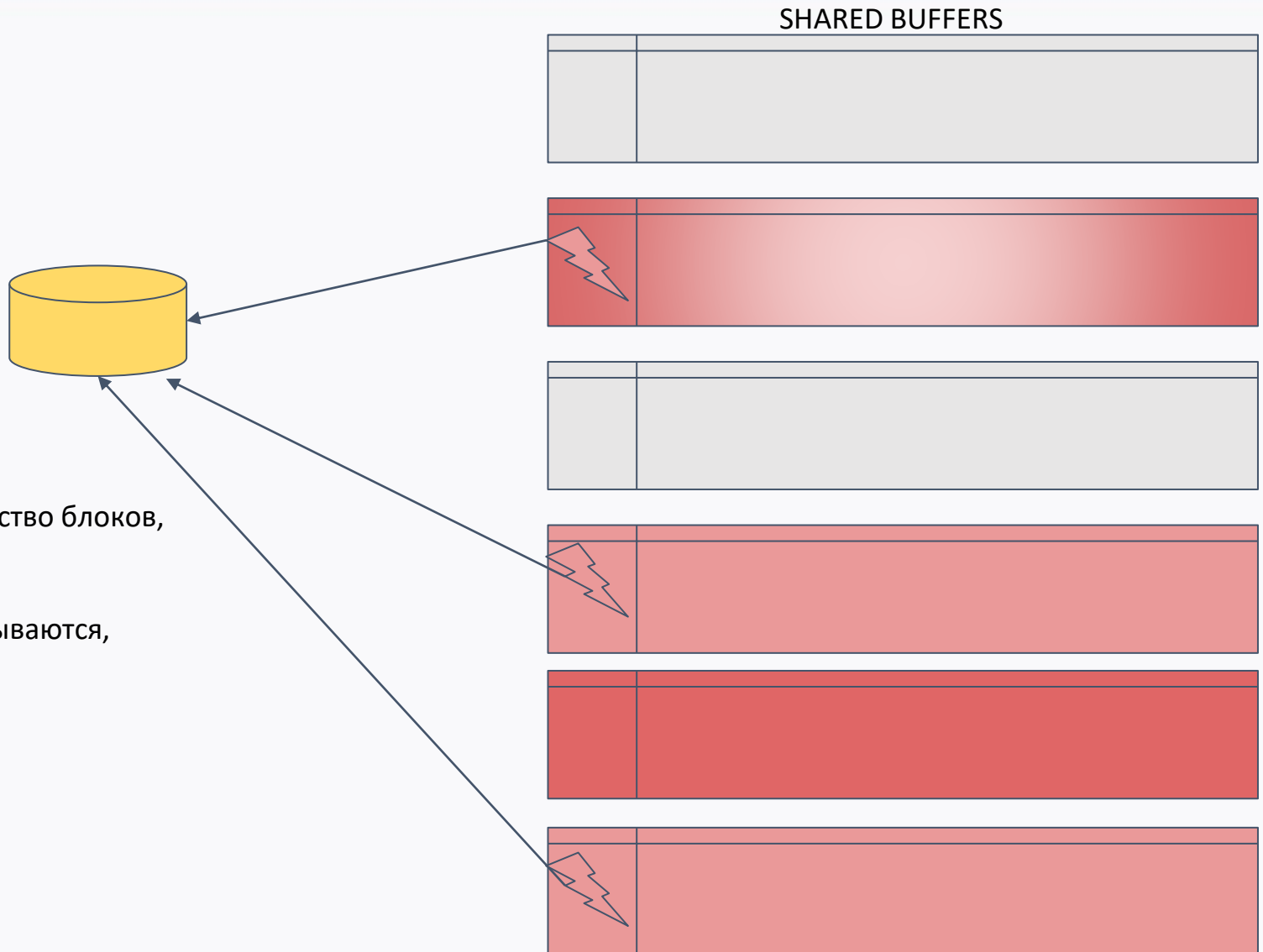
1. сначала сбрасываются буферы ХАСТ
2. помечаются грязные буферы

SHARED BUFFERS



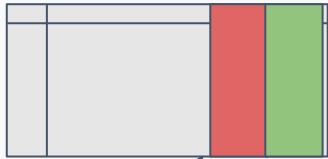
Контрольная точка

3. зная значение параметра `checkpoint_completion_target = 0.5` и количество блоков, которые нужно записать мы рассчитываем равномерную запись
4. помеченные страницы постепенно записываются,
5. пометка убирается из заголовка буфера

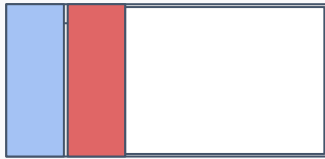


Контрольная точка

WAL



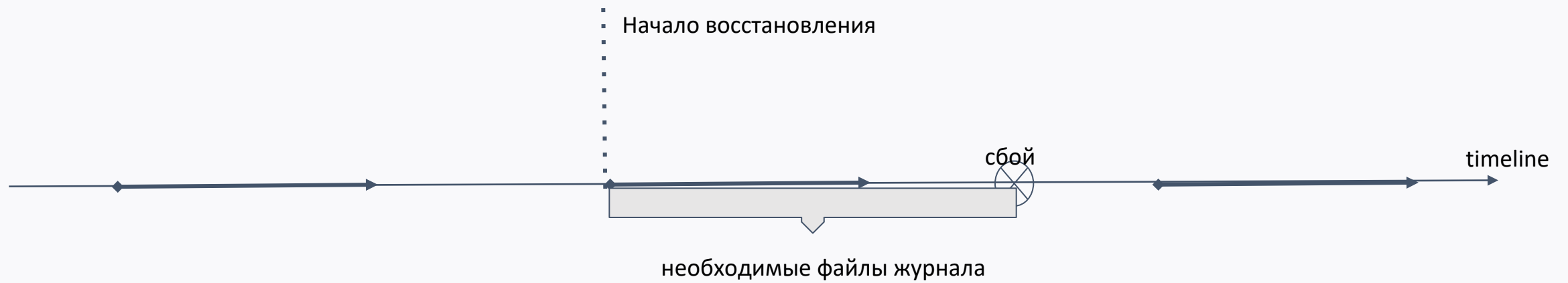
`pg_current_wal_lsn()`



запись о завершении КТ:
`latest_checkpoint_location: ____`

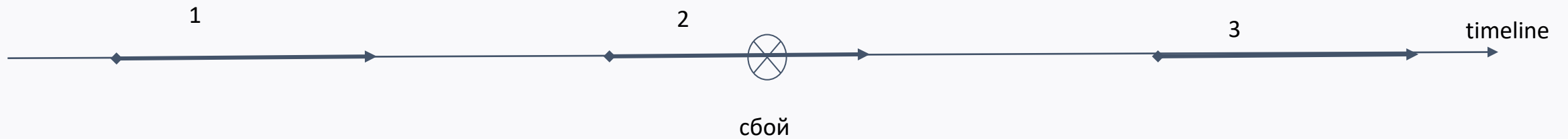
1. в журнале создается запись о завершении контрольной точки с указанием момента ее начала
1. в файл `$PGDATA/global/pg_control` записывается LSN контрольной точки
- ...
- Latest checkpoint location: 0/12B35EBB
- ...

Контрольная точка

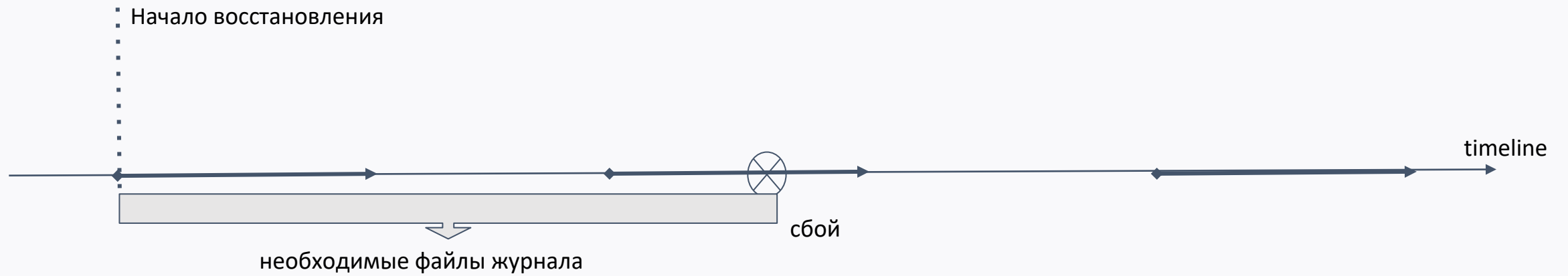


Контрольная точка

С какой точки произойдет восстановление и за какой период нужны будут wal файлы?



Контрольная точка



Контрольная точка

При старте сервера после сбоя

1. найти LSN0 начала последней завершенной контрольной точки
2. применить каждую запись журнала, начиная с LSN0 , если LSN записи больше, чем LSN страницы
3. перезаписать нежурналируемые таблицы init-файлами
4. выполнить контрольную точку

Контрольная точка

Настройка частоты срабатывания:

- `checkpoint_timeout = 5min`
- `max_wal_size = 1GB`

Сервер хранит журнальные файлы необходимые для восстановления:

- $(2 \text{ (1 с 12 версии)} + \text{checkpoint_completion_target}) * \text{max_wal_size}$
- еще не прочитанные через слоты репликации
- еще не записанные в архив, если настроена непрерывная архивация
- не превышающие по объему минимальной отметки

Настройки

- `max_wal_size = 1GB`
- `min_wal_size = 100MB`
- `wal_keep_segments = 0`

Контрольная точка. Процесс фоновой записи

в какой-то момент все свободные блоки заканчиваются.. Что делать дальше?.



Контрольная точка. Процесс фоновой записи

Настройки

- `bgwriter_delay = 200ms`
- `bgwriter_lru_maxpages = 100`
`bgwriter_lru_multiplier = 2.0`

Алгоритм

- уснуть на `bgwriter_delay`
- если в среднем за цикл запрашивается N буферов, то записать $N * \text{bgwriter_lru_multiplier} \leq \text{bgwriter_lru_maxpages}$ грязных буферов

Контрольная точка. Практика

The image features a high-angle, aerial view of a dense urban skyline, likely New York City, with numerous skyscrapers and buildings. The entire image is overlaid with a semi-transparent blue and green gradient. A network of thin, light blue lines connects various points across the image, creating a digital or technological feel. The word "Вопросы?" is centered in the middle of the image in a large, white, sans-serif font.

Вопросы?

The background of the entire image is an aerial photograph of a dense city skyline, likely New York City, with numerous skyscrapers. A semi-transparent blue overlay covers the entire image. In the center, there is a horizontal band with a gradient from teal on the left to dark blue on the right. Overlaid on this band is a white network pattern of dots and lines. The title text is centered within this band.

Настрока журнала

Уровни журнала

Minimal

восстановление после сбоя

Replica

восстановление из резервной копии, репликация
+ операции массовой обработки данных, блокировки

Logical

логическая репликация
+ информация для логического декодирования

Настройка

wal_level = replica

Настройка записи на диск

Синхронизация с диском

данные должны дойти до энергонезависимого хранилища через многочисленные кэши

СУБД сообщает операционной системе способом, указанным в `wal_sync_method`
надо учитывать аппаратное кэширование

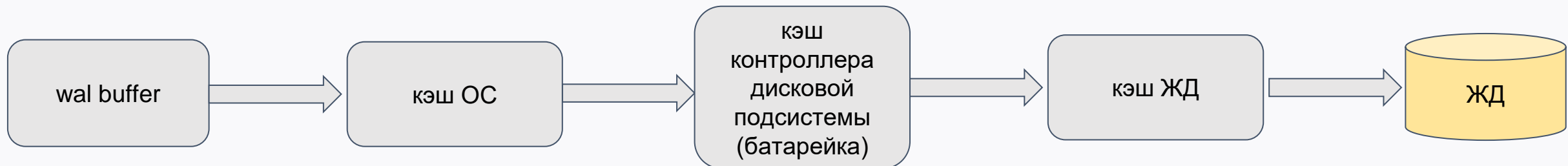
Настройки

`fsync = on`

`show fsync;`

`show wal_sync_method;`

утилита `pg_test_fsync` помогает выбрать оптимальный способ



Повреждение данных

Контрольные суммы журнальных записей

включены всегда, CRC-32

Контрольные суммы страниц (накладные расходы)

По умолчанию отключены. До 12 версии можно включить только при инициализации кластера.

`pg_createcluster --data-checksums`

Настройки

`show data_checksums;`

`ignore_checksum_failure = off`

`wal_log_hints = off` (записывает все содержимое каждой страницы при изменениях даже инф.бит, неявно on при контрольных суммах страниц)

`wal_compression = off`

<https://postgrespro.ru/docs/postgrespro/13/app-pgchecksums>

Характер нагрузки

Постоянный поток записи

- характер нагрузки отличается от остальной системы
- последовательная запись, отсутствие случайного доступа
- при высокой нагрузке — размещение на отдельных физических дисках (символьная ссылка из `$PGDATA/pg_wal`)

Редкое чтение

- при восстановлении
- при работе процессов `walsender`, если реплика не успевает быстро получать записи

Режимы записи

- **синхронный режим**
- **асинхронный режим**

Режим синхронной записи

Алгоритм

- при фиксации изменений сбрасывает накопившиеся записи, включая запись о фиксации
- ждет `commit_delay`, если активно не менее `commit_siblings` транзакций

Характеристики

- гарантируется долговечность
- увеличивается время отклика

Настройки

- `synchronous_commit = on`
- `commit_delay = 0`
- `commit_siblings = 5`

Режим асинхронной записи

Алгоритм

- циклы записи через `wal_writer_delay`
- записывает только целиком заполненные страницы
- но если новых полных страниц нет, то записывает последнюю до конца

Характеристики

- гарантируется согласованность, но не долговечность
- зафиксированные изменения могут пропасть ($3 \times \text{wal_writer_delay}$)
- уменьшается время отклика

Настройки


- `synchronous_commit = off` (можно изменять на уровне транзакции)
- `wal_writer_delay = 200ms`

The image features a high-angle, aerial view of a dense urban skyline, likely New York City, with numerous skyscrapers and buildings. The entire image is overlaid with a semi-transparent blue layer. A network of thin, light blue lines connects various points across the blue area, creating a digital or technological aesthetic. The Russian word "Порефлексируем" is centered in the middle of the image in a large, white, sans-serif font.

Порефлексируем

Вопросы?


- Кто что запомнил за сегодня?
- Сколько контрольных точек рекомендовано хранить для гарантированного восстановления?
- Как вам баланс между теорией и практикой?

The image features a high-angle, aerial view of a dense urban landscape, likely New York City, with numerous skyscrapers and buildings. The entire image is overlaid with a semi-transparent blue and green gradient. A network of thin, light blue lines connects various points across the image, creating a digital or data network aesthetic. In the center, the Cyrillic letters 'ДЗ' are prominently displayed in white.

ДЗ

ДЗ

1. Настройте выполнение контрольной точки раз в 30 секунд.
2. 10 минут с помощью утилиты `rgbench` подавайте нагрузку.
3. Измерьте, какой объем журнальных файлов был сгенерирован за это время. Оцените, какой объем приходится в среднем на одну контрольную точку.
4. Проверьте данные статистики: все ли контрольные точки выполнялись точно по расписанию. Почему так произошло?
5. Сравните `tps` в синхронном/асинхронном режиме утилитой `rgbench`. Объясните полученный результат.
6. Создайте новый кластер с включенной контрольной суммой страниц. Создайте таблицу. Вставьте несколько значений. Выключите кластер. Измените пару байт в таблице. Включите кластер и сделайте выборку из таблицы. Что и почему произошло? как проигнорировать ошибку и продолжить работу?



Заполните, пожалуйста,
опрос о занятии по ссылке в чате
<https://otus.ru/polls/39605/>

The background of the entire slide is an aerial photograph of a dense city skyline, likely New York City, with numerous skyscrapers. A semi-transparent blue overlay covers the image, featuring a white geometric network pattern of interconnected dots and lines. The text is centered in the middle of the slide.

Спасибо за внимание!
Приходите на следующие вебинары

Аристов Евгений
Золотов Антон