

Тема 3: Архитектура: буферный кеш и журнал



PostgreSQL

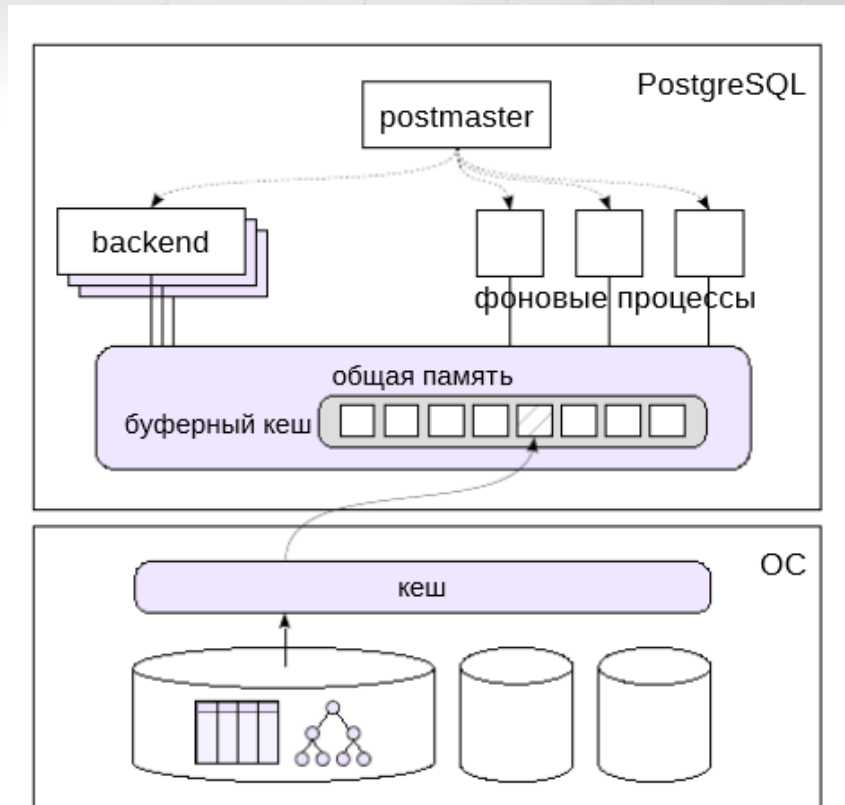
- 1) Устройство буферного кеша**
- 2) Алгоритм вытеснения**
- 3) Журнал предзаписи**
- 4) Контрольная точка**

Массив буферов

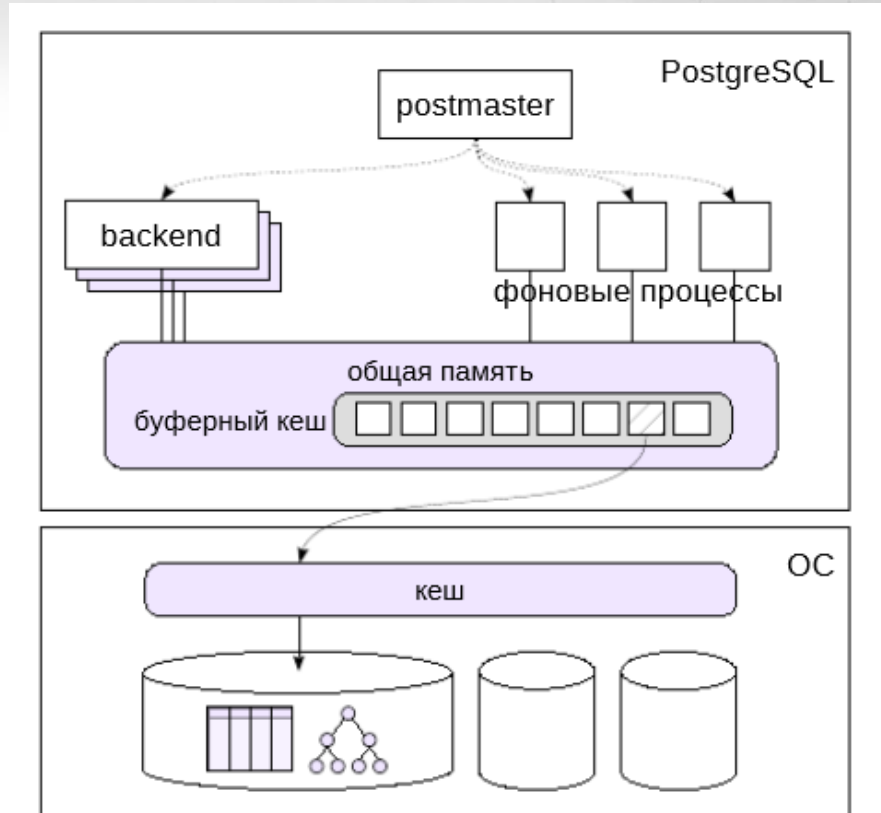
- страница данных (8 Кбайт)
- доп. информация

Блокировки в памяти

- для совместного доступа



**Вытеснение редко
используемых страниц**
- «грязный» буфер
записывается на диск
- на освободившееся место
читается другая страница



Проблема: при сбое теряются данные из оперативной памяти, не записанные на диск

Журнал

**поток информации о выполняемых действиях,
позволяющий повторно выполнить потерянные при сбое операции
запись попадает на диск раньше, чем измененные данные**

Журнал защищает

**страницы таблиц, индексов и других объектов
статус транзакций (хаст)**

Журнал не защищает

временные и нежурналируемые таблицы

Периодический сброс всех грязных буферов на диск

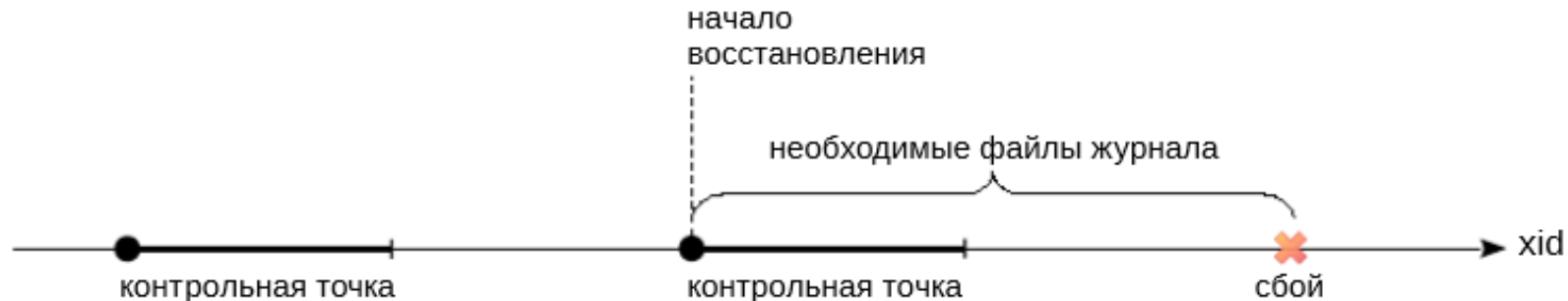
гарантирует попадание на диск всех изменений до контрольной точки

ограничивает размер журнала, необходимого для восстановления

Восстановление при сбое

начинается с последней контрольной точки

последовательно проигрываются записи, если изменений нет на диске

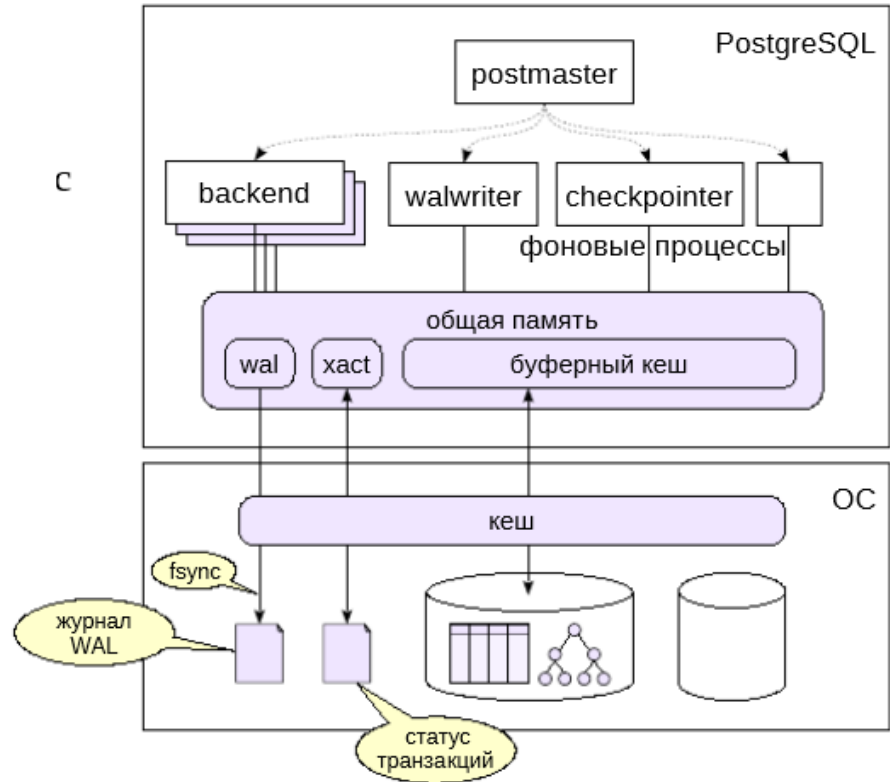


Синхронный режим

- запись при фиксации;
- обслуживающий процесс.

Асинхронный режим

- фоновая запись;
- walwriter.



Буферный кеш существенно ускоряет работу, уменьшая число дисковых операций

Надежность обеспечивается журналированием

Размер журнала ограничен благодаря контрольным точкам

Журнал удобен и используется во многих случаях:

- для восстановления после сбоя;**
- при резервном копировании;**
- для репликации между серверами.**

1. Проверьте, как используется буферный кеш в случае обновления одной строки в обычной и во временной таблице.

Попробуйте объяснить отличие.

2. Создайте нежурналируемую таблицу и вставьте в нее несколько строк. Сымитируйте сбой системы, остановив сервер в режиме `immediate`, как в демонстрации.

Запустите сервер и проверьте, что произошло с таблицей.

Найдите в журнале сообщений сервера упоминание о восстановлении после сбоя.

1. Временные таблицы выглядят так же, как обычные, но время их жизни — текущий сеанс. Такая таблица видна тоже только в текущем сеансе.

Воспользуйтесь командой

EXPLAIN (analyze, buffers, costs off, timing off)

как было показано в демонстрации.

2. Останов в режиме immediate выполняется так:

sudo pg_ctlcluster 12 main stop -m immediate --skip-systemctl-redirect

Ключ **--skip-systemctl-redirect** нужен здесь из-за того, что используется PostgreSQL, установленный в Ubuntu из пакета. Он управляется командой **pg_ctlcluster**, которая вызывает утилиту **systemctl**, которая в свою очередь вызывает **pg_ctl**, теряя при этом указанный режим. Ключ позволяет обойтись без **systemctl** и передать информацию **pg_ctl**.