Журналирование
Журнал предзаписи

Авторские права
© Postgres Professional, 2019 год.
Авторы: Егор Рогов, Павел Лузанов

Использование материалов курса
Некоммерческое использование материалов курса (презентации, демонстрации) разрешается без ограничений. Коммерческое использование возможно только с письменного разрешения компании Postgres Professional. Запрещается внесение изменений в материалы курса.

Обратная связь
Отзывы, замечания и предложения направляйте по адресу:
edu@postgrespro.ru

Отказ от ответственности
Компания Postgres Professional не несет никакой ответственности за любые повреждения и убытки, включая потерю дохода, нанесенные прямым или непрямым, специальным или случайным использованием материалов курса. Компания Postgres Professional не предоставляет каких-либо гарантий на материалы курса. Материалы курса предоставляются на основе принципа «как есть» и компания Postgres Professional не обязана предоставлять сопровождение, поддержку, обновления, расширения и изменения.

Темы

Журнал упреждающей записи (WAL)

Логическое и физическое устройство журнала

Процесс упреждающей записи и восстановление

Журнал

Postgres PROFESSIONAL

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

Основная причина существования журнала — необходимость восстановления согласованности данных в случае сбоя, при котором теряется содержимое оперативной памяти, в частности, буферный кэш.

Тем самым обеспечивается выполнение свойства долговечности (буква «D» из набора свойств транзакций ACID).

Одновременно с изменением данных в странице буферного кэша в журнале создается запись, содержащая достаточную информацию для повторения этой операции. Журнальная запись в обязательном порядке попадает на диск (или другое энергонезависимое устройство) до того, как туда попадет измененная страница — отсюда и название: «журнал предзаписи», «write-ahead log».

В случае сбоя можно прочитать журнал и при необходимости повторить те операции, которые уже были выполнены, но результат которых не успел попасть на диск.

https://postgrespro.ru/docs/postgresql/10/wal-intro

Что защищено журналом

Изменение любых страниц в буферном кэше

- в том числе страницы таблиц и индексов
- кроме нежурналируемых и временных таблиц

Фиксация и отмена транзакций — буферы XACT

Файловые операции

- создание и удаление файлов
- создание и удаление каталогов

Журналировать нужно все операции, при выполнении которых возможна ситуация, что при сбое результат операции не дойдет до диска.

В частности, в журнал записываются следующие действия:

- изменение страниц в буферном кэше (как правило, это страницы таблиц и индексов) — так как измененная страница попадает на диск не сразу;

- фиксация и отмена транзакций — точно так же, изменение статуса происходит в буфере XACT и попадает на диск не сразу;

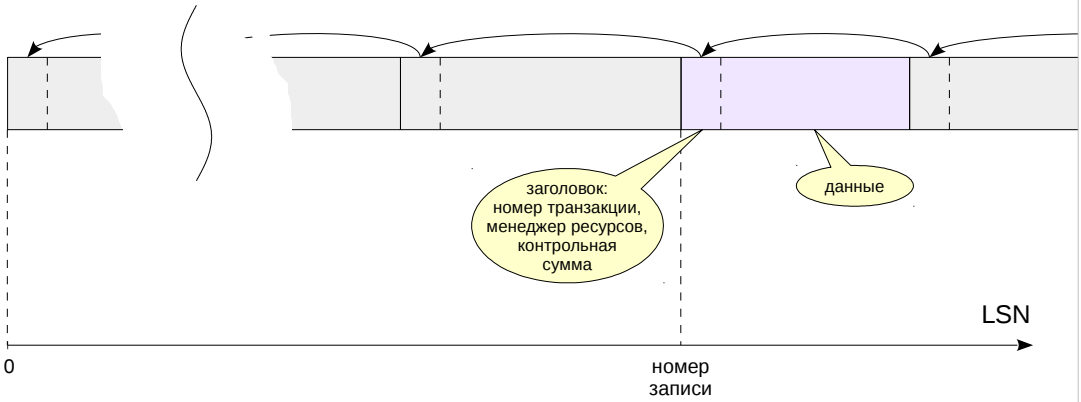
- файловые операции (создание и удаление файлов и каталогов, например, создание файлов при создании таблицы) — так как эти операции должны происходить синхронно с изменением данных.

При этом в журнал не записываются:

- операции с нежурналируемыми (и временными) таблицами — их название говорит само за себя;

До версии PostgreSQL 10 не журналировались хеш-индексы (они служили только для сопоставления функций хеширования различным типам данных), но сейчас это исправлено.

Логическое устройство



последовательность записей

номер записи — 64-битный LSN (log sequence number)

специальный тип pg_lsn

5

Логически журнал можно представить себе как последовательность записей различной длины. Каждая запись содержит данные о некоторой операции, предваренные заголовком. В заголовке, в числе прочего, указаны:

- номер транзакции, к которой относится запись;

- менеджер ресурсов — компонент системы, ответственный за данную запись;

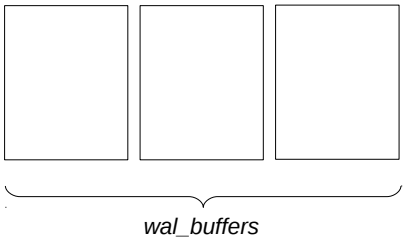
- контрольная сумма (CRC).

Сами данные могут иметь разный смысл. Менеджер ресурсов «понимает», как интерпретировать данные в своей записи. Есть отдельные менеджеры для таблиц, для каждого типа индекса, для статуса транзакций и т. п. Например, данные могут представлять собой некоторый фрагмент страницы, который надо записать поверх ее содержимого с определенным смещением.

Для того, чтобы сослаться на определенную запись, используется тип данных pg_lsn (LSN = log sequence number) — 64-битное число, представляющее собой байтовое смещение до записи относительно начала журнала.

https://postgrespro.ru/docs/postgresql/10/datatype-pg-lsn

Физическое устройство



В памяти

кольцевой буферный кэш

wal_buffers = –1

1/32 *shared_buffers*

На диске

файлы (сегменты) по 16 МБ

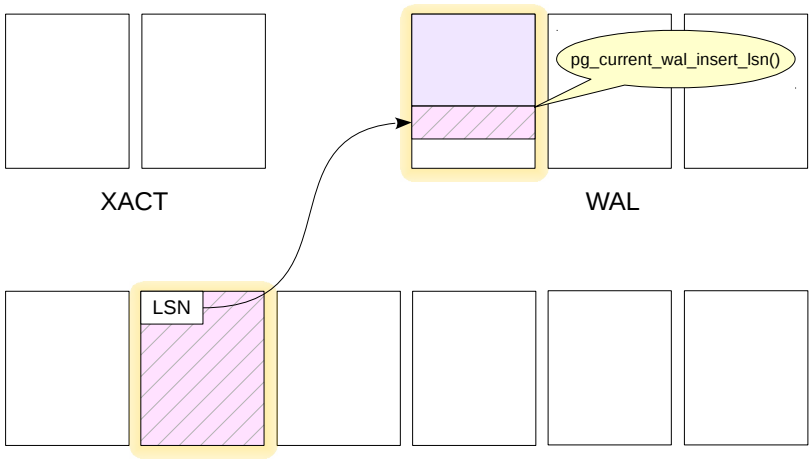
На диске журнал хранится в виде файлов в каталоге $PGDATA/pg_wal. Каждый файл занимает 16 МБ. Размер может быть изменен только при компиляции исходных кодов (начиная с PostgreSQL 11 размер можно будет изменять при инициализации кластера).

Журнальные записи попадают в текущий использующийся файл; когда он заканчивается — начинает использоваться следующий.

В оперативной памяти для журнала выделены специальные буферы. Размер кэша задается параметром *wal_buffers* (значение по умолчанию подразумевает автоматическую настройку: выделяется 1/32 часть буферного кэша).

Журнальный кэш устроен наподобие буферного кэша, но работает преимущественно в режиме кольцевого буфера: записи добавляются в «голову» буфера, а записываются на диск с «хвоста».

Упреждающая запись



Проиллюстрируем сказанное выше про упреждающую запись.
На слайде показаны три важные области общей памяти экземпляра:

- буферный кэш (размером *shared_buffers*),

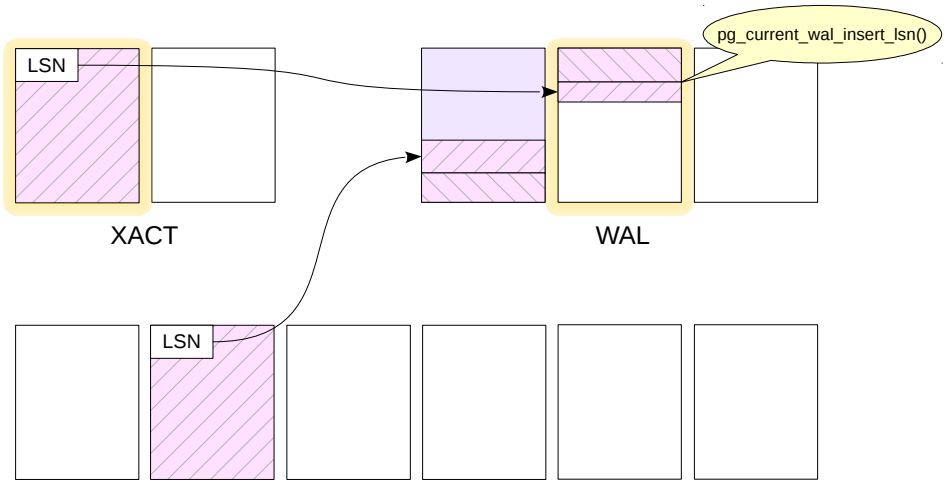
- только что рассмотренный журнальный кэш WAL (размером *wal_buffers*),

- кэш состояния транзакций, называемый также XACT (размером 128 страниц).

При изменении страницы данных в буферном кэше, формируется журнальная запись. Она помещается в страницу журнала, а ссылка на запись помещается в специальном поле LSN в заголовке страницы данных.

Позицию для записи можно узнать с помощью функции pg_current_wal_insert_lsn().

Упреждающая запись

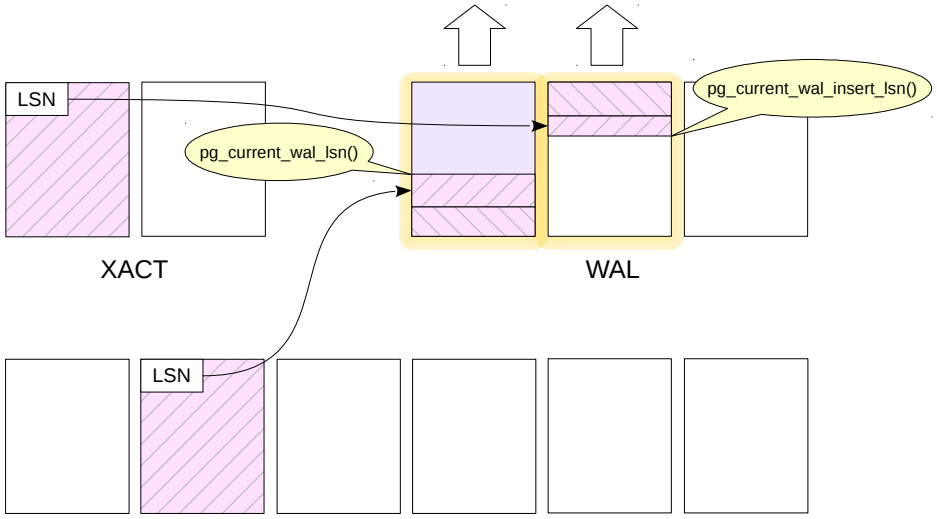


Допустим, далее происходит фиксация транзакции. Для этого формируется журнальная запись, меняется бит состояния на странице XACT и ссылка на эту запись проставляется в поле LSN измененной страницы.

При вставке указатель pg_current_wal_insert_lsn сдвигается вперед.

Заметим, что между записями, относящимися к одной транзакции, могут попасть записи других транзакций, относящихся к любой БД. Журнал — общий для всего кластера.

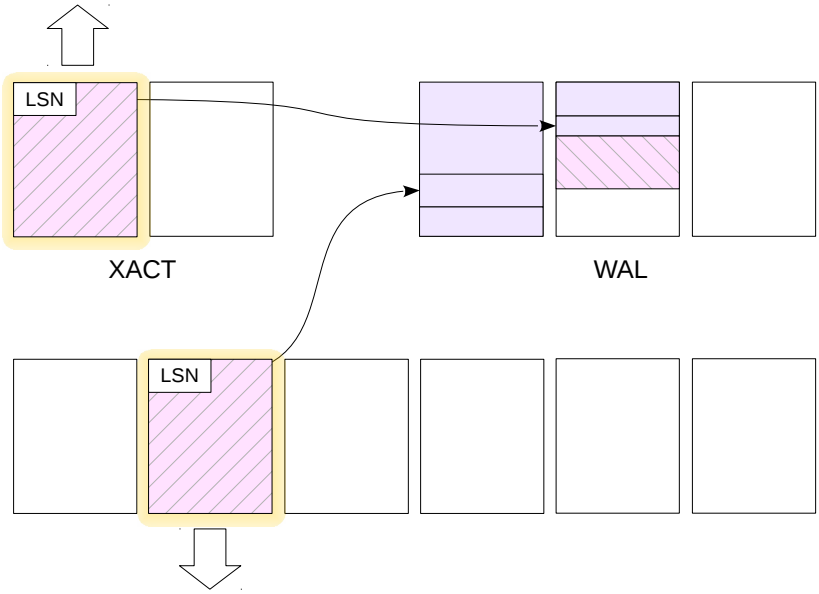
Упреждающая запись



Далее в какой-то момент (в какой именно — будет рассмотрено в теме «Настройка журнала») журнальные записи, которые еще не попали на диск, должны на него попасть.

Функция pg_current_wal_lsn() показывает последнюю запись, уже дошедшую до диска.

Упреждающая запись



Только после того, как на диск попали журнальные записи, могут быть записаны и сами измененные страницы. Порядок контролируется с учетом LSN последнего изменения страницы и текущего состояния pg_current_wal_lsn. При этом работа продолжается, в журнал будут попадать новые и новые записи. Главное, чтобы запись с LSN последнего изменения страницы была на диске.

Если окажется, что страница данных должна быть записана (например, она вытесняется из буферного кэша), а журнальная запись еще не попала на диск, журнальные буферы сбрасываются принудительно.

Восстановление

Алгоритм (упрощенный)

при старте сервера после сбоя
(состояние кластера в `pg_control` отличается от «shut down»):

1. для каждой журнальной записи:
   1.1. определить страницу, к которой относится эта запись
   1.2. применить запись, если ее LSN больше, чем LSN страницы
2. перезаписать нежурналируемые таблицы init-файлами

Если в работе сервера произошел сбой, то при последующем запуске процесс startup (запускаемый postmaster-ом в самом начале работы) обнаружит это, посмотрев в файл pg_control и увидев статус, отличный от «shut down». Тогда автоматически будет выполнено восстановление.

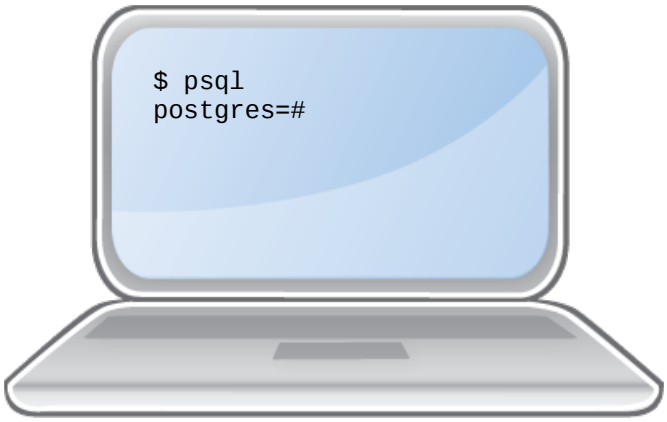
Процесс startup будет последовательно читать журнал и применять записи к страницам, если в этом есть необходимость (что можно проверить, сравнив LSN страницы на диске с LSN журнальной записи). Изменение страниц происходит в буферном кэше, как при обычной работе — для этого postmaster запускает необходимые фоновые процессы.

Аналогично записи применяются и к файлам: например, если запись говорит о том, что файл должен существовать, а его нет — файл создается.

В конце процесса все нежурналируемые таблицы перезаписываются с помощью образов в init-файлах.

Приведенный алгоритм является упрощенным. В частности, ничего не говорится о том, с какого места надо начинать чтение журнальных записей (это будет рассмотрено в теме «Контрольная точка»).

Демонстрация



Итоги

Использование буферов в оперативной памяти
приводит к необходимости журналирования

Журнал содержит информацию,
позволяющую повторно выполнить операции после сбоя
и восстановить согласованность

Журнал всегда записывается на диск до того,
как записываются измененные страницы данных

Практика

1. Создайте таблицу с первичным ключом и добавьте в нее несколько строк.
2. Сколько байтов занимают сгенерированные журнальные записи?
   Чем можно объяснить довольно большое число?
3. Посмотрите заголовки этих журнальных записей утилитой pg_waldump и проверьте предположения, сделанные в п. 2.